

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

Curso de Graduação em Sistemas de Informação

Ricardo Vitor Santos Hamada

**SISTEMA DE ORGANIZAÇÃO DE OBJETOS DE APRENDIZAGEM NO FORMATO
IEEE/LOM ATRAVÉS DE VÍDEOS DO YOUTUBE**

Diamantina

2022

Ricardo Vitor Santos Hamada

**SISTEMA DE ORGANIZAÇÃO DE OBJETOS DE APRENDIZAGEM NO FORMATO
IEEE/LOM ATRAVÉS DE VÍDEOS DO YOUTUBE**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação, como parte dos requisitos exigidos para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Alessandro Vivas Andrade

**Diamantina
2022**



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

FOLHA DE APROVAÇÃO

Ricardo Vitor Santos Hamada

SISTEMA DE CRIAÇÃO DE OBJETOS DE APRENDIZAGEM NO FORMATO IEEE/LOM ATRAVÉS DE VÍDEOS DO YOUTUBE

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação pela Universidade Federal dos Vales do Jequitinhonha e Mucuri.

Aprovada em 10/01/2023

BANCA EXAMINADORA

Prof. Alessandro Vivas Andrade
Faculdade de Ciências Exatas - UFVJM

Prof. Eduardo Pelli
Faculdade de Ciências Exatas - UFVJM

Prof. Rafael Santin
Faculdade de Ciências Exatas - UFVJM



Documento assinado eletronicamente por **Alessandro Vivas Andrade, Servidor (a)**, em 26/01/2023, às 18:17, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rafael Santin, Servidor (a)**, em 27/01/2023, às 14:46, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Eduardo Pelli, Servidor (a)**, em 27/01/2023, às 17:04, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufvjm.edu.br/sei/controlador_externo.php?aa=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0957638** e o código CRC **CB3E6C69**.

RESUMO

Os avanços tecnológicos permitiram a inserção de ferramentas para dar suporte aos conteúdos pedagógicos que fomentam o processo de aprendizagem. Uma dessas ferramentas criadas para fazer a construção do conhecimento e tornar o processo de aprendizagem mais assertivo foi o chamado Objeto de Aprendizagem, doravante (OA). Com base nessas informações, este trabalho propõe um Sistema *Web* que utiliza a plataforma do YouTube, também conhecido por ser um repositório de OA baseado em vídeos. O sistema tem por objetivo retornar, salvar e organizar os OAs no formato IEEE/LOM, que é um padrão usado para classificar os Objetos com a finalidade de localizá-los pelos mecanismos de busca; e posteriormente utilizá-los nos demais contextos com vistas à aprendizagem. Os OAs obtidos através dos dados dos vídeos providos da *Application Programming Interface* (API) do YouTube, permitirão ao usuário criar um repositório de OA com as informações dos vídeos salvos, bem como editar e excluir as informações que foram salvas no banco de dados. Pensando nisso, o sistema foi desenvolvido com base na linguagem de programação *Python* pela sintaxe simples e eficiência em manipulação de dados. Para tanto, foram utilizados o *Micro-Framework Flask* por possuir uma estrutura descomplicada e entendível a qual permite que um projeto possua apenas os recursos necessários para funcionar. O banco de dados utilizado foi o MongoDB por ser específico para armazenar dados em JSON que é o formato retornado pela API do YouTube, além de utilizar a própria API que serviu para extrair os OAs. Como resultado obteve-se um Objeto de Aprendizagem estruturado no padrão IEEE/LOM chegando ao objetivo final do sistema.

Palavras-chave: API. Dados. Informação. Repositório.

ABSTRACT

Technological advancements have allowed for the inclusion of tools to support pedagogical content that promotes the learning process. One such tool created to build knowledge and make the learning process more effective is called a Learning Object (LO). Based on this information, this work proposes a Web System that uses the YouTube platform, also known as a video-based LO repository. The system's goal is to retrieve, save and organize LOs in the IEEE/LOM format, which is a standard used to classify objects for search engine purposes, and then use them in other contexts for learning. The LOs obtained from video data from the YouTube Application Programming Interface (API) will allow the user to create an LO repository with the saved video information, as well as edit and delete information saved in the database. With this in mind, the system was developed using the Python programming language for its simple syntax and data manipulation efficiency. The Micro-Framework Flask was used for its uncomplicated and understandable structure, allowing the project to have only the necessary features to function. The database used was MongoDB, as it is specific for storing data in JSON format, which is the format returned by the YouTube API, besides using the API itself to extract the LOs. The result was a structured LO in the IEEE/LOM standard, achieving the final goal of the system.

Keywords: API. Data. Information. Repository.

LISTA DE ILUSTRAÇÕES

Figura 1 – Front-end e Back-end	17
Figura 2 – Dados disponíveis na API do YouTube	23
Figura 3 – Operações Compatíveis com a API do YouTube	23
Figura 4 – Escopo do padrão LOM	32
Figura 5 – Comparativo entre os padrões de metadados	34
Figura 6 – Visão geral de funcionamento do sistema	37
Figura 7 – Caso de Uso do Sistema (Ampliação da imagem encontra-se em Anexo A) .	38
Figura 8 – Fluxograma de Processos do Sistema (Ampliação da imagem encontra-se em Anexo B)	39
Figura 9 – Tela Principal do Sistema	40
Figura 10 – Termo a ser pesquisado	40
Figura 11 – Resultados da pesquisa	41
Figura 12 – Botões Visualizar e Salvar	41
Figura 13 – Página de Objetos de Aprendizagem Salvos	42
Figura 14 – Visualização do vídeo no YouTube	42
Figura 15 – Antes da ação de deletar	43
Figura 16 – Depois da ação de deletar	43
Figura 17 – Campos para edição	44
Figura 18 – Botões cancelar e salvar edição	44
Figura 19 – Metadados do vídeo	45
Figura 20 – Botões Voltar e Editar	45
Figura 21 – Categorias de Metadados IEEE LOM	47
Figura 22 – Caso de Uso do Sistema Ampliado	53
Figura 23 – Fluxograma de Processos do Sistema Ampliado	55

LISTA DE ABREVIATURAS E SIGLAS

ABED	Associação Brasileira de Educação a Distância
ADL	Advanced Distributed Learning
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
CESTA	Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem
CSS	Cascading Style Sheet
GPL	General Public License
HTML	Hyper Text Markup Language
IBM	International Business Machines
IDE	Ambientes de Desenvolvimento Integrado
IEEE	Institute of Electrical and Electronic Engineers
IEEE LTSC	Learning Technology Standards Committee
LOM	Learning Object Metadata
MR	Modelo Relacional
NoSQL	Not only SQL
OA	Objeto de Aprendizagem
OBAA	Objetos de Aprendizagem Baseados em Agentes
PSF	Python Software Foundation
RDF	Resource Description Framework
REST	Representational state transfer
RIVED	Rede Interativa Virtual de Educação
SCORM	Sharable Content Object Reference Model
SGBD	Sistema Gerenciador de Banco de Dados
TIC	Tecnologias da Informação

UFRGS	Universidade Federal do Rio Grande do Sul
UNISINOS	Universidade do Vale dos Sinos
UoD	Universo de Discurso
WSGI	Web Server Gateway Interface
WWW	World Wide Web

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo	16
1.2	Estrutura do Estudo	16
2	DESENVOLVIMENTO <i>WEB</i>	17
2.1	<i>A Web</i>	17
2.2	<i>Front-end e Back-end</i>	17
2.2.1	<i>Front-end</i>	18
2.2.2	<i>Back-end</i>	18
2.2.2.1	<i>Banco de Dados</i>	19
2.2.2.1.1	Banco de Dados Relacional	19
2.2.2.1.2	Banco de Dados Não Relacional (NoSQL)	20
2.3	<i>Frameworks</i>	20
2.3.1	<i>Micro-Framework</i>	20
2.3.1.1	<i>Flask</i>	21
2.4	<i>API</i>	21
2.4.1	<i>APIs REST</i>	21
2.4.1.1	<i>API YouTube</i>	22
2.4.1.1.1	Recursos e tipos de recursos	22
2.4.1.1.2	Operações Compatíveis	23
3	<i>PYTHON</i>	25
3.1	Características	25
3.2	Histórico	25
3.3	Ferramentas de Desenvolvimento	25
3.3.1	<i>Editores</i>	26
3.3.2	<i>Ambientes de Desenvolvimento Integrado (IDE)</i>	26
3.3.3	<i>Depuradores</i>	26
3.3.4	<i>Profilers</i>	26
3.3.5	<i>Bibliotecas</i>	27
4	OBJETO DE APRENDIZAGEM	29
4.1	Definição	29
4.2	Repositórios de Objetos de Aprendizagem	29
4.3	Metadados	30
4.3.1	<i>Tipos de Metadados</i>	30
4.3.2	<i>Padrões de Metadados</i>	30
4.3.2.1	<i>O padrão Duplin Core (DC)</i>	30

4.3.2.2	<i>O padrão OBAA</i>	31
4.3.2.3	<i>O padrão IEEE LOM</i>	31
4.3.2.4	<i>O padrão IEEE SLT</i>	32
4.3.2.5	<i>O padrão SCORM</i>	33
4.3.2.6	<i>O padrão Experience API</i>	33
4.3.2.7	<i>Comparativo dos padrões de metadados</i>	34
5	METODOLOGIA	35
5.1	Objetivo Geral	35
5.1.1	<i>Objetivos Específicos</i>	36
5.2	Desenvolvimento do Sistema	36
5.3	Visão geral do Sistema	37
5.4	Funcionamento do Sistema	38
5.4.1	<i>Caso de Uso do Sistema</i>	38
5.4.2	<i>Fluxograma do Sistema</i>	39
5.4.3	<i>Telas do Sistema</i>	40
6	RESULTADOS ALCANÇADOS	47
6.1	Objeto de Aprendizagem gerado pelo sistema	48
7	CONSIDERAÇÕES FINAIS	49
	REFERÊNCIAS	51
	ANEXO A – FIGURA 7 AMPLIADA	53
	ANEXO B – FIGURA 8 AMPLIADA	55
	ANEXO C – OBJETO DE APRENDIZAGEM GERADO PELO SISTEMA	57

1 INTRODUÇÃO

Na última década, as tecnologias têm contribuído para um impacto significativo na sociedade, no que se refere a pesquisas. No contexto informacional, por exemplo, observa-se um crescimento exponencial no consumo de conteúdo, o que antes era encontrado por meio de leitura em revistas, enciclopédias e jornais, agora pode ser facilmente obtido através de dispositivos eletrônicos com acesso à *Internet*. No âmbito educacional, as tecnologias de informação tornaram o processo de aprendizagem mais eficiente e rápido, ao oferecer uma ampla variedade de plataformas e recursos que podem ser facilmente acessados, contribuindo assim para diferentes formas de apoio à aprendizagem.

Na questão educacional, os avanços tecnológicos permitiram a inserção de ferramentas que dão suporte aos conteúdos pedagógicos os quais fomentam o processo de aprendizagem do aluno. Um exemplo disso é o uso dos Objetos de Aprendizagem, recursos digitais reutilizáveis que podem ser usados para dar suporte ao ensino e ao aprendizado. Embora não exista uma definição precisa para os OA, estes geralmente incluem materiais como vídeos, áudios, imagens, animações e outros tipos de conteúdo digital que podem ser utilizados para apoiar o ensino. Por isso, são considerados como uma forma eficaz de personalizar a aprendizagem e tornar o processo mais atrativo e interativo para os alunos.

Com o objetivo de facilitar o modo de selecionar e recuperar as informações, os Objetos de Aprendizagem são organizados em repositórios. Um repositório, no contexto de OAs, é um local de armazenamento especificamente projetado para armazenar materiais educacionais para fins de ensino e aprendizagem. Esse termo se refere a qualquer espaço físico ou digital onde os objetos podem ser armazenados e acessados conforme necessário. Os repositórios são uma ferramenta essencial na organização e manutenção dos OAs, pois garantem que sejam facilmente recuperáveis e prontos para uso em ambientes educacionais. Segundo Pinheiro (2022), repositórios são como “depósitos virtuais” que armazenam e gerenciam os OAs, permitindo que professores e estudantes encontrem e utilizem os recursos de aprendizagem de maneira eficiente.

Com base nos conceitos apresentados, este trabalho propõe um Sistema *Web* que utiliza a plataforma do YouTube, também conhecido por ser um repositório de Objetos de Aprendizagem, baseado em vídeos. O objetivo deste sistema é buscar, salvar e organizar os OAs em formato IEEE/LOM, um padrão amplamente utilizado para classificar e encontrar objetos, para que possam ser utilizados em contextos de aprendizagem. Os OAs serão obtidos através da API do YouTube, permitindo que o usuário crie um repositório personalizado com as informações dos vídeos. Além de buscar informações através de termos específicos, o sistema permite ainda salvar os OAs em um banco de dados e dá ao usuário a liberdade de editar ou excluir esses dados a qualquer momento.

O sistema foi desenvolvido com base na linguagem de programação *Python* pela sintaxe simples e eficiência em manipulação de dados, para isso, foram utilizados o *Micro-Framework Flask* por possuir uma estrutura descomplicada e entendível a qual permite que

um projeto possua apenas os recursos necessários para funcionar, o banco de dados utilizado foi o MongoDB por ser específico para armazenar dados em JSON. Este é o formato retornado pela API do YouTube, que além de utilizar a própria API serviu para extrair os OAs. Como resultado, obteve-se um Objeto de Aprendizagem estruturado no padrão IEEE/LOM chegando, então, ao objetivo final do sistema.

A revisão de literatura abordou conceitos que fizeram parte da construção do sistema como, por exemplo, o Desenvolvimento *Web* e suas ferramentas, a linguagem *Python*, os Objetos de Aprendizagem e o padrão IEEE/LOM. Após a revisão, apresentou-se a metodologia realizada para criar o sistema e como foi feito o desenvolvimento do mesmo. Por fim, foram apresentados os resultados gerados pelo sistema e as considerações finais do trabalho.

1.1 Objetivo

Este trabalho tem como objetivo desenvolver um sistema para extrair, organizar e armazenar Objetos de Aprendizagem (OAs) a partir de vídeos do YouTube, seguindo o padrão IEEE/LOM. Além disso, o sistema permitirá aos usuários editar e excluir OAs. A criação deste sistema visa oferecer acesso fácil a informações relevantes e organizadas para o processo de aprendizagem, tornando mais simples a busca e utilização desses objetos.

1.2 Estrutura do Estudo

A estrutura do trabalho se organiza como segue. A Seção 2 demonstra as tecnologias utilizadas no desenvolvimento *Web* e que serviram de pilares para a criação do sistema do trabalho. Nessa Seção são abordados conceitos como o que é a *Web*, *Front-end*, *Back-end*, banco de dados, *Frameworks* e *API's*.

A Seção 3 aborda sobre a linguagem de programação utilizada no sistema, o *Python*, bem como suas características, histórico e ferramentas de desenvolvimento.

A Seção 4 define o conceito dos Objetos de Aprendizagem e contextualiza sua importância para o estudo, além de tratar sobre os repositórios dos OAs existentes, o conceito de metadados, os padrões de metadados existentes e um comparativo entre eles.

A Seção 5 apresenta o sistema desenvolvido e discute sobre seus objetivos, como foi feita sua construção e o seu funcionamento.

A Seção 6 apresenta os resultados alcançados através do uso do sistema além do Objeto de Aprendizagem organizado na estrutura do padrão IEEE/LOM.

A Seção 7 apresenta as considerações finais, algumas ressalvas sobre o sistema e sugestões para trabalhos futuros.

2 DESENVOLVIMENTO *WEB*

O Desenvolvimento *Web* representa a codificação de páginas para a *Internet*, ou seja, nada mais é que a criação de *sites*, páginas, portais e aplicativos para *Web*. No entanto, essa codificação vai além da utilização do HTML e CSS, que são ferramentas essenciais para criação de uma página. O Desenvolvimento *Web* é responsável por toda a estruturação de um *site*, tanto da parte na qual o usuário consegue visualizar e interagir (*Front-end*), quanto a parte por trás do *site*, que é a responsável por comunicar com o servidor (*Back-end*), e que promove a experiência ao utilizar a aplicação. Primeiro, antes de falar sobre os conceitos relacionados ao Desenvolvimento *Web*, precisa-se entender o que é a *Web*.

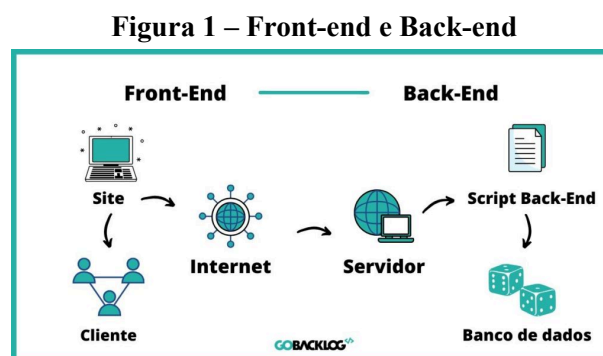
2.1 A *Web*

A *Web* (*World Wide Web*, ou *WWW*), de acordo com Latorre (2018), pode ser entendida como um sistema de links de hipertexto que tem por objetivo conectar documentos (*sites*), que estão disponíveis na *Internet* e que se comunicam através da tecnologia digital. O “hipertexto” é considerado a mistura de arquivos de vários formatos e tipos (textos, imagens, vídeos, gráficos) em um mesmo documento.

A *Web* não pode ser confundida com a *Internet*: a *Web* é um subconjunto da *Internet* que dá acesso a informações através de um navegador. Todos os sites, como e-mails, o *Facebook*, o Wikipédia, a *Amazon*, dentre outros, fazem parte da *Internet*, mas não da *Web*.

2.2 *Front-end e Back-end*

O Desenvolvimento *Web* é dividido em duas partes fundamentais, o *Front-end* e o *Back-end*. O *Front-end* é a parte responsável pela interface visual e interativa, que é acessada pelo usuário. O *Back-end* é a parte responsável pelo funcionamento da aplicação, incluindo a comunicação com o servidor, banco de dados, entre outras funcionalidades. A Figura 1 faz uma ilustração do *Front-end* e o *Back-end*.



Fonte: ANTUNES, 2022

2.2.1 *Front-end*

O *Front-end* é a parte responsável pela interface visual e interativa de um *site*, ou seja, é a parte que o usuário pode interagir. A estruturação da interface do *Front-end* geralmente utiliza três tecnologias básicas: HTML (*Hyper Text Markup Language*), CSS (*Cascading Style Sheet*) e JavaScript. O HTML é usado para estruturar o conteúdo da página, o CSS é usado para definir a aparência e o *layout*, e o JavaScript é usado para adicionar interatividade e dinâmica à página. Essas tecnologias trabalham juntas para criar uma experiência enriquecedora e funcional para o usuário.

- HTML: Do inglês *Hyper Text Markup Language*, é uma linguagem de marcação utilizada para estruturar os elementos de uma página *Web*, como parágrafos, links, títulos, tabelas, imagens e vídeos. Ela permite que os desenvolvedores adicionem estrutura e significado ao conteúdo de uma página, através de *tags* que são marcas específicas que indicam ao navegador de que forma os elementos devem ser exibidos. O HTML não é uma linguagem de programação propriamente dita, ele fornece apenas a estrutura básica para que outras tecnologias, como CSS e JavaScript, possam ser usadas para criar uma experiência enriquecedora ao usuário;
- CSS: *Cascading Style Sheet*, o CSS é uma linguagem usada para definir o estilo visual de uma página web, como o tamanho, cor, fonte e posicionamento dos elementos. Ele não é considerado uma linguagem de programação, mas sim uma linguagem de estilos, pois permite aos desenvolvedores criar designs atraentes e consistentes, sem precisar modificar diretamente o código HTML. O CSS é usado para separar a estrutura e conteúdo da aparência de uma página, tornando mais fácil de manter e modificar o design de um *site*;
- JavaScript: O JavaScript é uma linguagem de programação que é usada para adicionar interatividade e dinâmica às páginas *Web*. Ele é usado para criar eventos, como cliques de botão, aparição de telas, movimentos na página, controle de multimídia e imagens animadas. O JavaScript é executado no lado do cliente, ou seja, no navegador do usuário, e permite que as páginas sejam mais interativas e reajam às ações do usuário, sem precisar ser atualizadas pelo servidor. Ele é uma das três tecnologias fundamentais do desenvolvimento *Web*, junto com HTML e CSS, e é usado para complementar e aprimorar a estrutura e a estética das páginas *Web*.

2.2.2 *Back-end*

O *Back-end* é a parte invisível de uma aplicação *Web*, responsável por controlar e processar as informações que são apresentadas ao usuário. Ele é responsável por gerenciar as conexões com os servidores e bancos de dados, processar as solicitações dos usuários, realizar cálculos e lógicas, e gerar as respostas que são exibidas no *Front-end*. O *Back-end* também é responsável pela criação de APIs (*Application Programming Interface*) que permitem a comuni-

cação entre diferentes sistemas e plataformas, e pela implementação das regras de negócio que regem o funcionamento do *site*. O *Back-end* é essencial para garantir que a aplicação funcione de maneira eficiente e segura, e é geralmente escrito em linguagens de programação como *Java*, *Python* ou *Ruby*.

Para entender melhor o funcionamento do *Back-end* é necessário abordar o conceito de Banco de Dados.

2.2.2.1 Banco de Dados

O banco de dados é uma estrutura organizada de dados armazenados em um sistema de computador, geralmente usando modelos de dados relacionais ou não relacionais. Ele é projetado para armazenar e gerenciar grandes quantidades de dados de forma eficiente e segura, e permite acesso rápido e preciso a esses dados através de consultas *Structured Query Language* (SQL) ou outras linguagens de consulta. Os dados que compõem o banco de dados são frequentemente coletados de fontes relacionadas ao mundo real, como sistemas de informação empresariais, e são geralmente normalizados para garantir a integridade dos dados e evitar redundância.

2.2.2.1.1 Banco de Dados Relacional

O modelo relacional é uma abordagem para organizar informações em um banco de dados, relacionando-os entre si através de tabelas estruturadas. Segundo Pintar (2016), essa estruturação permite uma organização lógica e fácil para acessar os dados, uma vez que cada tabela tem sua própria estrutura de colunas e linhas. Os dados são organizados em forma de tabelas, as quais, individualmente, cada tabela representa uma entidade ou conceito específico; e cada linha representa um registro específico dessa entidade. A estrutura de colunas permite a identificação de diferentes atributos ou características relacionadas a essa entidade. O modelo relacional permite estabelecer relações entre tabelas, permitindo acesso a informações relacionadas de forma rápida e precisa.

Uma característica importante que define o modelo relacional é o uso de chaves primárias e chaves estrangeiras. As primárias servem como identificadores únicos para cada registro em uma tabela, garantindo que não haja duplicidade de dados. As estrangeiras estabelecem relações entre registros em tabelas diferentes, permitindo acesso a informações que estejam relacionadas de forma precisa e eficiente. Elas permitem e garantem a integridade dos dados, evitando dados redundantes e garantindo a consistência destes ao longo do tempo. Além disso, essas chaves permitem a realização de operações como atualizações, exclusões e junções de tabelas.

2.2.2.1.2 Banco de Dados Não Relacional (NoSQL)

Segundo LÓSCIO, OLIVEIRA e PONTES (2011), o NoSQL (*Not only SQL*) é uma abordagem de banco de dados que surgiu para atender às necessidades de gerenciamento de grandes volumes de dados não estruturados e semiestruturados, bem como a escalabilidade. Ele permite a manipulação de dados não estruturados, sem a necessidade de seguir as regras rígidas do modelo relacional. Além disso, o NoSQL é projetado para escalar horizontalmente, permitindo que mais recursos sejam adicionados, conforme o aumento da necessidade de armazenamento e processamento de dados. Isso o torna uma solução ideal para aplicações *Web* de grande escala, *Big Data*, IoT e outros cenários em que se faz necessário lidar com grandes quantidades de dados.

A grande diferença do NoSQL para o SQL é que o primeiro foi criado para ser escalável, além de possuir grande performance, isso se deve ao fato de o NoSQL conseguir armazenar uma quantidade massiva de dados sem ter que dividir esses registros em tabelas, linhas e colunas como é feito no SQL.

Para Cardoso (2012) o NoSQL é um bancos de dados desenvolvido para ser de fácil distribuição. Por sua vez, Brito (2010) descreve que o modelo NoSQL é dividido em quatro categorias básicas:

- “Armazenamento chave-valor (Amazon Dynamo)” (BRITO, 2010);
- “Orientados a documentos (CouchDB e o MongoDB)” (BRITO, 2010);
- “Orientados a coluna (Cassandra e o HyperTable)” (BRITO, 2010);
- “Orientados a grafos (Neo4j e o ArangoDB)” (BRITO, 2010).

2.3 Frameworks

Os *Frameworks* são conceitos importantes no Desenvolvimento *Web*, considerados códigos prontos que fornecem alguma funcionalidade específica e ajudam no desenvolvimento do *software*. De acordo com Barro (2022), o *Framework* pode ser visto como uma estrutura pré-construída a qual facilita o processo de criação de um sistema ou aplicação, podendo ser considerado um modelo que tem como objetivo principal trazer agilidade no desenvolvimento sem que seja necessário iniciar a construção do sistema do início.

2.3.1 Micro-Framework

Os *Micro-Frameworks* podem ser considerados *Frameworks* com estruturas simples que fornecem apenas o básico para o projeto entrar em funcionamento, com objetivo de permitir que o desenvolvedor incremente as funcionalidades ao longo do projeto, permitindo que o mesmo não contenha recursos inutilizáveis, ou desnecessários para a aplicação. Um exemplo de *Micro-Framework* é o *Flask*, que foi utilizado no desenvolvimento do sistema proposto no trabalho.

2.3.1.1 *Flask*

O *Flask* é um *Micro-Framework* muito utilizado para criar aplicações *Web*. Para tanto, é implementada apenas a funcionalidade principal dando liberdade aos desenvolvedores de adicionarem os recursos necessários durante a implementação. Esse *Micro-Framework* pode ser usado tanto para *back-end* quanto para *front-end*. O Werkzeug e Jinja são as duas bibliotecas principais do *Flask*.

Além disso, o *Flask* fornece um servidor WSGI (*Web Server Gateway Interface*) para desenvolvimento local, incluindo o cliente de teste para simular as solicitações HTTP. O WSGI é um padrão do *Python* que faz a especificação de interface e descreve como um servidor da *Web* comunica com os aplicativos *Web*.

Conforme mencionado, o Jinja é uma dependência do *Flask*, um modelo completo de *template engine* que permite criar páginas HTML para as aplicações *Python*. É dito *template engine* porque o Jinja possibilita incorporar o código do *Python* nas páginas HTML fazendo com que haja uma comunicação entre a página e a aplicação de modo mais fácil e sem complicações. Além disso, o código escrito no modelo HTML é compilado como código *Python*.

Como o *Flask* é considerado um *Micro-Framework* ele possui várias extensões que podem ser adicionadas ao longo do projeto para fornecer mais recursos à aplicação, algumas delas podem ser usadas para comunicação com o banco de dados (*SQLAlchemy*), criação de formulários (*Flask-WTF*), controle de acesso, cache, envio de *e-mails* e demais funções.

2.4 API

As APIs (*Application Programming Interface* ou *Interface* de Programação de Aplicação) são estruturas que permitem que duas aplicações se comuniquem entre si através de um conjunto de padrões e protocolos. O objetivo das APIs é permitir incrementar novas funcionalidades ao *software*, além de reduzir o tempo de desenvolvimento que seria usado para criar essas funcionalidades. As APIs usam o esquema de solicitações e respostas para realizar a comunicação entre dois sistemas.

As documentações das APIs possuem todas as informações necessárias sobre como são feitas as estruturas de solicitações e respostas da comunicação da API.

2.4.1 *APIs REST*

As APIs REST são as mais conhecidas e encontradas na *Web*. Seu funcionamento se deve ao envio de solicitações do cliente ao servidor. O servidor por sua vez responde as solicitações com dados para o cliente. O REST são funções que os clientes usam para acessar os dados do servidor, como exemplo tem-se as funções *GET* (Ler), *POST* (Criar), *PUT* (Atualizar) e *DELETE* (Deletar) as quais permitem aos clientes fazerem as solicitações que eles desejarem.

A API REST possui uma característica que é considerada a sua principal que é a ausência de estado, que significa que os servidores não realizam o salvamento dos dados das solicitações que os clientes fazem através da API, os dados são retornados pelos servidores, mas não salvos.

Um exemplo de API que pode ser considerada uma API REST é a API do YouTube.

2.4.1.1 API YouTube

O YouTube disponibiliza as funções executadas em sua plataforma através da API de dados, permitindo que outras aplicações ou sites façam uso dessas funcionalidades.

A API suporta métodos para inserir, atualizar ou excluir os recursos disponíveis do YouTube. Um recurso é representado por uma experiência do YouTube que pode ser um vídeo, uma inscrição, uma *playlist* de vídeos (YOUTUBE, 2019).

Para fazer o uso da API é necessário seguir os seguintes requisitos, de acordo com o YouTube (2019):

1. Uma chave de API deve ser especificada para cada solicitação feita ou ser fornecido um *token* do OAuth 2.0;
2. Para cada solicitação de inserção, atualização e exclusão é necessário enviar um *token* de autorização;
3. Na API alguns métodos para recuperação de recursos podem conter parâmetros que precisam de autorização.

2.4.1.1.1 Recursos e tipos de recursos

“Um recurso é uma entidade de dados que possui uma identificação exclusiva” (YOUTUBE, 2019). A Figura 2, a seguir, descreve os diferentes tipos de recursos com os quais o usuário pode interagir usando a API.

Figura 2 – Dados disponíveis na API do YouTube

Recursos	
activity	Contém informações sobre uma ação que determinado usuário executou no site do YouTube. Ações do usuário que são informadas em feeds de atividades incluem a classificação de um vídeo, o compartilhamento de um vídeo, a marcação de um vídeo como favorito, a publicação de um boletim do canal etc.
channel	Contém informações sobre um canal simples do YouTube.
channelBanner	Identifica o URL que será usado para definir uma imagem recém-enviada como imagem do banner de um canal.
guideCategory	Identifica uma categoria que o YouTube associa aos canais com base em seu conteúdo ou outros indicadores, como a popularidade. As categorias de guia servem para organizar canais de modo que os usuários do YouTube possam encontrar com mais facilidade o conteúdo que procuram. Embora os canais possam ser associados a uma ou mais categorias de guia, não é certeza que eles estejam em uma delas.
playlist	Representa uma playlist simples do YouTube. Uma playlist é um conjunto de vídeos que podem ser visualizados em sequência e compartilhados com outros usuários.
playlistItem	Identifica um recurso, como um vídeo, que faz parte de uma playlist. O recurso playlistItem também contém detalhes que explicam como o recurso incluso é usado na playlist.
search result	Contém informações sobre um vídeo, um canal ou uma playlist do YouTube que corresponde aos parâmetros de pesquisa especificados em uma solicitação da API. Embora indique um recurso exclusivamente identificável (como um vídeo), um resultado de pesquisa não tem seus próprios dados persistentes.
subscription	Contém informações sobre a inscrição de um usuário do YouTube. Uma assinatura notifica o usuário quando novos vídeos são adicionados a um canal ou quando outro usuário executa uma das várias ações no YouTube, como o upload ou a classificação de um vídeo ou comentários sobre um vídeo.
thumbnail	Identifica imagens em miniatura associadas a um recurso.
video	Representa um vídeo simples do YouTube.
videoCategory	Identifica uma categoria que foi ou pode ser associada a vídeos enviados.

Fonte: YOUTUBE, 2019

2.4.1.1.2 Operações Compatíveis

A Figura 3 mostra os métodos mais comuns compatíveis com a API do YouTube.

Figura 3 – Operações Compatíveis com a API do YouTube

Operações	
list	Recupera (GET) uma lista vazia ou com recursos.
insert	Cria (POST) um novo recurso.
update	Modifica (PUT) um recurso existente para propagar os dados em sua solicitação.
delete	Remove (DELETE) um recurso específico.

Fonte: YOUTUBE, 2019

3 PYTHON

3.1 Características

De acordo com Borges (2014), o *Python* possui uma sintaxe simples, que permite que o código fonte seja bem legível, fazendo com que a linguagem seja bastante produtiva e fácil de aprender. Nesse mesmo ano, Borges cita uma outra característica interessante do *Python* que é o código portátil, com o qual é possível compilar suas aplicações em um sistema e depois rodar em outras plataformas.

Isso acontece porque a linguagem é interpretada pela sua própria máquina virtual através de *bytecode*. O *bytecode* é o formato intermediário entre o código fonte escrito pelo desenvolvedor e o código da máquina executado pelo computador. É importante mencionar também que o *Python* é um *software* de código aberto, o que contribui para sua incorporação em diversos produtos e *software* de várias empresas.

3.2 Histórico

Segundo Borges (2014), o *Python* foi criado em 1990 por Guido van Rossum; e provindo de outra linguagem chamada ABC. Hoje, a linguagem é usada por várias empresas de tecnologia, tais como Google, Microsoft, Disney.

De acordo com Montoro (2012), o *Python* foi desenvolvido para possuir um código de fácil leitura, entendimento e aprendizado permitindo que funcionalidades simples e avançadas fizessem parte da estrutura da linguagem, de forma a englobar tanto desenvolvedores iniciantes quanto os que já possuem um conhecimento aprofundado no *Python*. Nas primeiras versões da linguagem as características principais foram a base para orientação a objetos, estruturas de dados de alto nível e tratamento de exceções.

Os direitos do *Python* pertencem a *Python Software Foundation* criada em março de 2001, sendo essa a instituição responsável pelo desenvolvimento de novas versões da linguagem, bem como a correção de erros e suporte.

3.3 Ferramentas de Desenvolvimento

Um dos fatores importantes a ser considerado no desenvolvimento de *software*, é o conjunto de ferramentas disponíveis com as quais se pode contar para ajudar no trabalho. Todas as linguagens possuem diferentes tipos de ferramentas de desenvolvimento de *software* que auxiliam no aumento da produtividade; e o *Python* não é uma exceção. Dentre essas ferramentas pode-se citar os editores, os Ambientes de Desenvolvimento Integrado (IDE), os depuradores, os *profilers* e as bibliotecas.

3.3.1 Editores

Os editores de texto são ferramentas usadas para criar e editar texto, porém podem ser usados para o desenvolvimento de *software*, pois permitem escrever o código-fonte e criar um arquivo a partir dele. Dentro deste grupo, existem muitos programas, desde os básicos como Bloco de notas, para os mais complexos, como *Sublime Text*, *Atom* e *Notepad++*.

Além disso, os editores de texto possuem várias funcionalidades que ajudam a melhorar a produtividade e que facilitam o desenvolvimento de *software*, como realce de sintaxe, correção gramatical e ortográfica, preenchimento automático, pesquisa usando expressões regulares, recuo automático, entre outros recursos.

3.3.2 Ambientes de Desenvolvimento Integrado (IDE)

Os Ambientes de Desenvolvimento Integrado são editores de texto evoluídos que estendem a funcionalidade dos editores adicionando ferramentas para facilitar a escrita do código e agilizar o processo de criação de *software*. Dentre essas ferramentas tem-se a depuração de código, o preenchimento automático, marcação de erro na sintaxe, entre outras. Alguns dos IDEs mais populares são o *Visual Studio*, o *Eclipse* e o *NetBeans*, em específico para o *Python* temos o *Pycharm*, o *Eric* e o *Wingware*.

3.3.3 Depuradores

Do inglês *debugging*, o processo de depuração de código é essencial para o desenvolvimento de *software* e para solucionar os problemas que causam o seu funcionamento, estes problemas ou erros são chamados de *bugs*. A depuração consiste em inspecionar o código passo a passo durante a execução do programa, permitindo que a parte causadora do erro seja encontrada. O depurador mais famoso para *Python* é o *pdb* que permite realizar o progresso passo a passo do programa, definir pontos de interrupção, além de poder avaliar os valores recebidos pelas variáveis ao longo da execução.

3.3.4 Profilers

Na engenharia de *software*, um *profiler* é uma ferramenta valiosa para medir e analisar o desempenho de um programa. Segundo Montoro (2012), um profiler é uma ferramenta que analisa o desempenho de outro programa, fornecendo estatísticas relevantes sobre sua execução. Essas informações incluem dados como o tempo de execução, o uso de memória, o número de chamadas de função e muito mais. Essas informações são fundamentais para identificar gargalos e pontos de melhoria no código, e são difíceis de obter sem o uso de um *profiler*, logo essa ferramenta é muito útil para melhorar o desempenho de um programa.

3.3.5 Bibliotecas

“As bibliotecas do *Python* são coleções de funções e módulos úteis para reduzir o uso de código no programa.” (HASHTAG, 2022).

A seguir, são citados alguns exemplos de bibliotecas *Python* de acordo com Hashtag (2022):

- Biblioteca *Requests*: É usada para permitir que um *site* execute interações com APIs;
- *FrameWork Flask*: É usado para o desenvolvimento de aplicações e *sites Web*, sendo uma ferramenta mais leve e intuitiva em comparação com o *Django*;
- *FrameWork Django*: É usado para o desenvolvimento de aplicações e *sites Web*, sendo uma ferramenta mais robusta e completa;
- Biblioteca NumPy e Biblioteca Pandas: São usadas para processamento de dados;
- Biblioteca OpenCV: É usada para processamento de imagens.

4 OBJETO DE APRENDIZAGEM

4.1 Definição

As Tecnologias de Informação transformaram a educação, fornecendo uma gama de ferramentas para o ensino-aprendizagem. Os Objetos de Aprendizagem (OA) fazem parte dessas ferramentas de destaque, pois facilitam o acesso a recursos valiosos que enriquecem e maximizam o processo de aprendizagem.

O conceito de Objeto de Aprendizagem foi definido por diversos autores ao longo do tempo, estes concordavam que um OA pode ser qualquer recurso que possa ser usado e reutilizado para contribuir ou atribuir a aprendizagem. O conceito de OA mais conhecido foi o atribuído por Wiley *et al.* (2000), definindo que um OA é “qualquer recurso digital que pode ser reusado para apoiar a aprendizagem”.

Autores como Tarouco *et al.* (2014) e Harman e Koohang (2007) seguiram a mesma linha de Wiley e compartilharam a ideia de que qualquer recurso, seja este digital ou não, que possa ser usado para apoiar a aprendizagem é considerado um Objeto de Aprendizagem.

4.2 Repositórios de Objetos de Aprendizagem

Um repositório pode ser entendido como um lugar, espaço ou depósito que pode ser utilizado para armazenar algo. No contexto dos Objetos de Aprendizagem isso não é diferente e serve justamente para armazenar recursos que possam contribuir para a aprendizagem, desses recursos podem-se mencionar imagens, textos, vídeos, dentre outros.

O Brasil não fica de fora no cenário de repositórios de Objetos de Aprendizagem, com o objetivo principal de fornecer uma estrutura eficaz para o suporte ao ensino e aprendizagem. Além disso, atende às necessidades dos educadores, oferecendo recursos úteis para serem utilizados em sala de aula.

De acordo com Souza¹, Albuquerque e Silva (2015), os Repositórios de Objetos Digitais de Aprendizagem existentes no Brasil são o Portal do Professor¹, RIVED² (Rede Interativa Virtual de Educação), o Banco Internacional de Objetos de Aprendizagem³, o Projeto Cesta⁴ (Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem), o Webeduc⁵. Contudo existem outros repositórios de OA no Brasil.

¹ O Portal do Professor está disponível em <<http://portaldoprofessor.mec.gov.br>>

² O RIVED está disponível em <<http://www.dmm.im.ufrj.br/projeto/rived/index.html>>

³ O Banco Internacional de Objetos de Aprendizagem está disponível em <<http://objetoseducacionais2.mec.gov.br/>>

⁴ O Projeto Cesta está disponível em <<http://www.cinted.ufrgs.br/CESTA/>>

⁵ O Webeduc está disponível em <<http://Webeduc.mec.gov.br/>>

4.3 Metadados

A palavra metadados é definida como dados que descrevem outros dados. Esses metadados fornecem informações estruturadas sobre um dado principal, tornando-as úteis e organizadas. Indo para um conceito mais técnico, Alves *et al.* (2010) afirmam que:

Os metadados são elementos descritivos ou atributos referenciais codificados que representam características próprias ou atribuídas às entidades; são ainda dados que descrevem outros dados em um sistema de informação, com o intuito de identificar de forma única uma entidade (recurso informacional) para posterior recuperação. (ALVES *et al.*, 2010).

Com base nessa definição, é possível notar que os metadados além de extraírem informações relevantes de um dado, possuem a característica de identificá-lo, singularmente, facilitando o processo de recuperação e reutilização em um sistema informacional.

4.3.1 Tipos de Metadados

De acordo com Riley (2017) existem quatro tipos de metadados, que são os:

- Descritivos - ajudam a identificar ou compreender um recurso;
- Administrativos – auxiliam na decodificação e renderização de arquivos, gerenciamento de arquivos de longo prazo e os direitos de propriedade intelectual associados ao conteúdo;
- Estruturais: servem para estabelecer as conexões entre as diferentes partes de um recurso;
- Linguagens de Marcação - servem para incorporar informações adicionais, como metadados, sinalizadores, e estruturas semânticas dentro do conteúdo.

4.3.2 Padrões de Metadados

Os padrões de metadados podem ser considerados como a estruturação para catalogar os metadados de acordo com suas características, possibilitando a identificação das informações e facilitando sua recuperação.

4.3.2.1 O padrão Duplin Core (DC)

O *Dublin Core* é um dos padrões de metadados que tem como objetivo simplificar a catalogação dos elementos digitais pela web e tem por finalidade facilitar a descrição de recursos como vídeos, sons, textos, imagens e páginas, além de ajudar na busca e recuperação desses recursos na web.

4.3.2.2 O padrão OBAA

O OBAA (Objetos de Aprendizagem Baseados em Agentes) é um padrão que utiliza como base a especificação IEEE LOM, ou seja, o conjunto de metadados do OBAA possui todas as categorias do IEEE LOM com o diferencial de ter os metadados de acessibilidade e segmentação adicionados como categorias. Uma das principais características do OBAA é que ele é multiplataforma, permitindo a distribuição e consumo das informações dos Objetos de Aprendizagem em plataformas como *mobile*, *tv*, *desktop*, *Web*, dentre outras.

4.3.2.3 O padrão IEEE LOM

O padrão LOM foi criado pelo *Institute of Electrical and Electronic Engineers* (IEEE) com o objetivo de classificar os Objetos de Aprendizagem para serem localizados pelos mecanismos de busca; e, posteriormente, serem utilizados nos demais contextos no que visa a aprendizagem.

Para caracterizar os OAs, o padrão LOM utiliza nove categorias: Geral, Ciclo de vida, Meta-metadados, Técnica, Educacional, Direitos, Relação, Anotação e Classificação.

Silva (2011) definiu as categorias da seguinte forma:

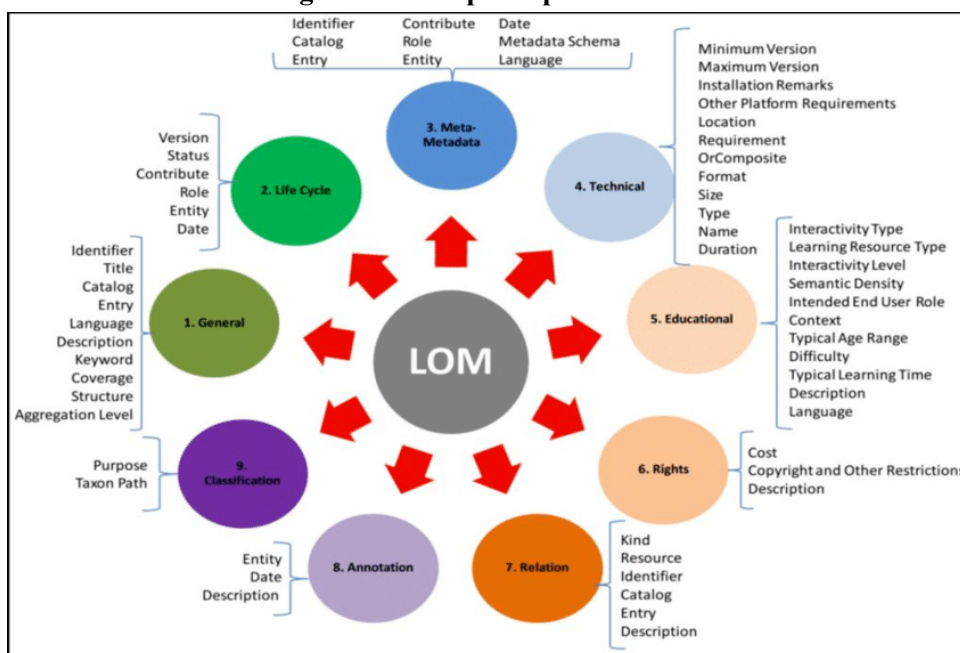
- Geral – incluem o nome, idioma, descrição e palavras-chave do OA, que são aspectos fundamentais para sua identificação e compreensão;
- Ciclo de vida – representa o estado atual do OA, bem como suas versões e histórico de evolução, fornecendo uma visão geral de sua evolução ao longo do tempo;
- Meta-metadados – são informações que descrevem o OA, fornecendo uma visão mais completa sobre suas características e utilidade;
- Técnica – as características técnicas do objeto de aprendizagem são enfatizadas, incluindo aspectos como o tamanho, formato, localização na Web, e a compatibilidade com navegadores e sistemas;
- Educacional – refere-se aos aspectos educacionais e pedagógicos do OA, que incluem o grau de interação, dificuldade, tipo de atividade e público-alvo;
- Direitos – fornece informações detalhadas sobre as condições de utilização e direitos de autor do OA, garantindo a proteção de seu conteúdo;
- Relação – estabelece conexões entre o OA e outros Objetos de Aprendizagem complementares ou similares, permitindo uma melhor integração e aproveitamento;
- Anotação – permite a realização de comentários sobre o OA, promovendo uma maior interação e colaboração entre os usuários;
- Classificação – permite a identificação do nível educacional do objeto, bem como suas características, incluindo pré-requisitos para utilização e classificação em termos de taxonomia, através de um sistema de classificação predefinido.

Pöttker, Ferneda e Moreiro-González (2018) enfatizam que:

O padrão LOM não interfere no conteúdo que compõe os Objetos de Aprendizagem porque ele apenas faz o agrupamento dos metadados através de um cabeçalho que fornece as informações sobre o Objeto de Aprendizagem, os elementos deste cabeçalho são metadados que descrevem semanticamente os objetos. Dessa forma, evidencia-se que o padrão LOM tem por objetivo facilitar a indexação, catalogação e busca de Objetos de Aprendizagem.(PÖTTKER; FERNEDA; MOREIRO-GONZÁLEZ, 2018).

A Figura 4 mostra a estrutura básica do padrão de metadados LOM.

Figura 4 – Escopo do padrão LOM



Fonte: PÖTTKER; FERNEDA; MOREIRO-GONZÁLEZ, 2018

4.3.2.4 O padrão IEEE SLT

Desenvolvida pelo comitê do *Institute of Electrical and Electronic Engineers* (IEEE) para tecnologias educacionais, a arquitetura (projeto IEEE SLT 11, 2002) foi criada com o objetivo de estruturar dados que retornam informações de alunos e seu contato com o recurso de aprendizagem. Esta arquitetura visa enfatizar os diferentes níveis do processo de aprendizagem, segundo Corbière e Choquet (2004):

1. O contexto para os atores humanos;
2. Um modelo de componentes para a organização;
3. O ciclo de desenvolvimento de *software*.

Corbière e Choquet (2004) ainda descrevem os cinco diferentes níveis da arquitetura que representam os diferentes pontos de vista de um processo de aprendizagem:

- Nível 1: Este nível é conceitualmente abstrato e envolve tarefas relacionadas à aquisição, transferência, intercâmbio e descoberta do conhecimento pelo aprendiz, decorrente de

suas interações com o ambiente. O ambiente e o aprendiz são concebidos como dois sistemas que trocam informações;

- Nível 2: Esta camada apresenta como o aluno responde ao seu ambiente, com um grau de detalhamento variável dentro do documento;
- Nível 3: Um sistema de componentes, normalizado pelo IEEE, define uma organização de um processo de aprendizagem visto do ponto de vista de fluxo de dados e controle;
- Nível 4: Este nível aborda diretamente o sistema de componentes a fim de estabelecer restrições tecnológicas de *design*. Ele permite a observação das atividades do sistema durante o processo de aprendizagem;
- Nível 5: Este nível se concentra na especificação das etapas gerais do ciclo de vida do *software*, tendo como base a arquitetura de componentes.

4.3.2.5 O padrão SCORM

O SCORM (*Sharable Content Object Reference Model*), em termos simples, é um padrão cujo principal objetivo é especificar como criar e compartilhar conteúdos de treinamento, ensino e educação entre sistemas de gerenciamento de aprendizagem. Os cursos compatíveis com SCORM alavancam o desenvolvimento do curso, garantindo que os cursos compatíveis sejam RAID, que segundo Jones (2022) significa:

- Reutilizável: pode ser adaptado e utilizado com diferentes ferramentas de desenvolvimento;
- Acessível: disponível para consulta e uso, conforme necessidade dos alunos e pelos desenvolvedores de conteúdo;
- Interoperável: compatível com uma vasta gama de dispositivos de *hardware*, sistemas operacionais e navegadores na *Web*;
- Durável: mantém sua funcionalidade sem necessidade de mudanças significativas ao longo do tempo, mesmo com a evolução do *software* do sistema.

4.3.2.6 O padrão Experience API

A *Experience API* (xAPI), é um padrão que faz o rastreamento, o registro e a coleta de dados do processo de experiência de aprendizado do usuário em sistemas de aprendizagem. Além disso, a xAPI tem por objetivo ajudar na interoperabilidade entre os sistemas, ou seja, melhorar a comunicação entre os sistemas de aprendizagem e o compartilhamento de dados em relação ao aprendizado dos alunos. A *Experience API* foi criada visando superar o SCORM, podendo ser considerada sua versão modernizada.

De acordo com Simão *et al.* (2018), a xAPI permite o armazenamento de dados que podem ser usados pelos professores para monitorar o desempenho dos estudantes, seja em

disciplinas gerais ou em habilidades específicas, tornando mais fácil sua avaliação e a avaliação do material de ensino.

4.3.2.7 Comparativo dos padrões de metadados

A Figura 5 demonstra um comparativo entre os padrões de metadados:

Figura 5 – Comparativo entre os padrões de metadados

Padrão	Instituição	Finalidade	Metadados
Dublin Core	DCMI	Facilitar a busca e recuperação de recursos baseados na Web	São definidos 15 elementos: título, criador, assunto, descrição, editor, colaborador, data, tipo, formato, identificador, fonte, linguagem, relação, cobertura e direitos
IEEE LOM	IEEE	Descrever objetos de aprendizagem e recursos digitais similares utilizados para suporte ao ensino e aprendizagem	Contém nove categorias (geral, ciclo de vida, meta-metadados, técnicos, educacionais, direitos, relação, anotação e classificação), que contém sub-elementos.
IEEE SLT	IEEE	Rastrear informações sobre as interações dos alunos com o conteúdo de aprendizagem	São descritos tipos de dados trocados: contexto de interação, observações, estado de aquisição, perfil do aluno, avaliação, referências do estudante, dados multimídia, localidade, conteúdos pedagógicos e consultas e informações catalogadas.
OBAA	UFRGS e Unisinos	Descrever objetos de aprendizagem, com foco na acessibilidade e na adaptabilidade na descrição de conteúdo de diferentes plataformas	Estende a IEEE LOM, adicionando 27 novos metadados na categoria 'técnicos', 8 na categoria 'educacionais', além de criar a categoria 'acessibilidade' (24 metadados) e de 'segmentação' (19 metadados)
SCORM	ADL	Desenvolvimento de materiais reutilizáveis, acessíveis, interoperáveis e duráveis para educação e treinamento	Aponta a utilização do padrão IEEE LOM para a descrição de seus componentes.
xAPI	ADL	Formalização, armazenamento e recuperação de eventos ligados à experiência de aprendizagem em ambientes virtuais	Os statements contêm 3 elementos principais, administrativos (id, timestamp, etc), descritivos (ator, verbo e objeto) e complementares (contexto).

Fonte: SIMÃO *et al.*, 2018

Com base no comparativo é possível extrair algumas informações dentre os padrões de metadados mostrados, sendo que três deles possuem a finalidade de descrever os objetos de aprendizagem, dos quais pode-se citar o OBAA, IEEE LOM e o Duplin Core. Já os padrões xAPI e IEEE SLT possuem em si a finalidade de registrar e descrever o contato dos estudantes com as ferramentas de aprendizagem, extraindo a experiência dos mesmos no processo. Por último, o SCORM é um padrão que foca em dispor um modelo para desenvolver conteúdos e treinamentos que visam o ensino e a educação.

5 METODOLOGIA

Inicialmente, foi realizada uma revisão da literatura para compreender os conceitos fundamentais relacionados ao Desenvolvimento *Web*, a linguagem de programação *Python* e Objetos de Aprendizagem. Esses conceitos foram de suma importância para a construção do sistema proposto no trabalho.

Em seguida, foi realizado um estudo detalhado da API do YouTube para identificar quais funcionalidades poderiam ser utilizadas para construir o sistema. A partir daí, observou-se que a API possui uma grande variedade de informações disponíveis, agrupadas em recursos. Para o desenvolvimento do sistema, foram selecionados os recursos de pesquisa e vídeos, já que permitem retornar os resultados de busca e informações sobre o vídeo escolhido pelo usuário.

Para implementação do sistema, foi utilizada a linguagem de programação *Python*, devido à sua sintaxe simples e eficiência na manipulação de dados, além de suas bibliotecas *redirect*¹, *request*², *pymongo*³ e JSON para realizar tarefas específicas. O Micro-Framework escolhido foi o Flask, pois possui um núcleo simples e permite que o projeto possua apenas os recursos necessários para funcionar, permitindo adicionar mais funcionalidades conforme a necessidade. O banco de dados utilizado foi o MongoDB⁴, por sua capacidade de armazenar grandes quantidades de dados em formato JSON e sua estrutura de documento, adequada para os dados retornados pela API. A interface do sistema foi baseada nas ferramentas HTML, CSS e JavaScript em conjunto com template engine Jinja2⁵.

Após o desenvolvimento, o sistema foi testado e validado com base nos objetivos propostos, avaliando a usabilidade e funcionalidade.

Por último, apresentou-se uma conclusão e algumas recomendações, para possíveis melhorias ou extensões do sistema.

Neste Capítulo é explicado os objetivos que levaram à criação do sistema, como foi realizado o desenvolvimento e o funcionamento, demonstrados, por meio de diagramas e telas.

5.1 Objetivo Geral

Extrair Objetos de Aprendizagem de vídeos do YouTube, organizá-los no padrão IEEE/LOM e realizar o armazenamento no banco de dados.

¹ A biblioteca *redirect* está disponível em <<https://flask.palletsprojects.com/en/2.2.x/quickstart/#redirects-and-errors>>

² A biblioteca *request* está disponível em <<https://flask.palletsprojects.com/en/2.2.x/api/#flask.request>>

³ A biblioteca *pymongo* está disponível em <<https://pypi.org/project/pymongo>>

⁴ O MongoDB está disponível em <<https://www.mongodb.com>>

⁵ O Jinja2 está disponível em <<https://palletsprojects.com/p/jinja>>

5.1.1 *Objetivos Específicos*

- Extrair metadados de vídeos do YouTube através de termos de busca, utilizando a API do YouTube;
- Fazer a organização das informações obtidas dos vídeos do YouTube em arquivos JSON⁶ (*JavaScript Object Notation*) baseados no padrão IEEE/LOM, que serão usados para criar repositórios de Objetos de Aprendizagem e armazená-los em um banco de dados;
- Fazer a gestão dos Objetos de Aprendizagem salvos no banco de dados permitindo editá-los e excluí-los.

5.2 **Desenvolvimento do Sistema**

O sistema foi desenvolvido com base na linguagem de programação *Python*, o motivo da escolha é a sintaxe simples e eficiência em manipulação de dados, além de possuir várias bibliotecas que proporcionaram a construção do sistema sem muitas complicações. As principais bibliotecas utilizadas foram a *redirect* que faz o redirecionamento do usuário para a página de destino, a *request* responsável por permitir a pesquisa dinâmica por meio das requisições HTTP enviadas para API do YouTube, a *pymongo* responsável por fazer a conexão do sistema com o banco de dados do MongoDB possibilitando o salvamento dos dados retornados da API e a *JSON* que permite a criação e leitura dos dados JSON retornados da API.

A ferramenta escolhida para construção do sistema foi o *Micro-Framework Flask*, o motivo da escolha é porque o *Flask* possui um núcleo simples e entendível que permite que um projeto possua apenas os recursos necessários para funcionar. Com isso, a medida que for precisando incrementar mais funções ao projeto é só ir importando as bibliotecas para fazerem a adição das funcionalidades.

Foi utilizada a própria API do YouTube para funcionamento da aplicação e organização dos Objetos de Aprendizagem, nela é possível fazer a escolha dos recursos que serão retornados como por exemplo os canais, comentários, as categorias, os vídeos, dentre outros. Para a aplicação foi escolhido o recurso de pesquisa da API que retorna os resultados da busca e o recurso de vídeos que retornam as informações do vídeo escolhido.

O banco de dados escolhido para salvar os dados da aplicação foi o MongoDB, porque o MongoDB é específico para armazenar dados em JSON que é o formato retornado pela API do YouTube, além de poder salvar grandes quantidades de dados em suas coleções (subpartes do banco geral) e em forma de documento o que difere do modelo relacional que armazena os registros em linhas e colunas.

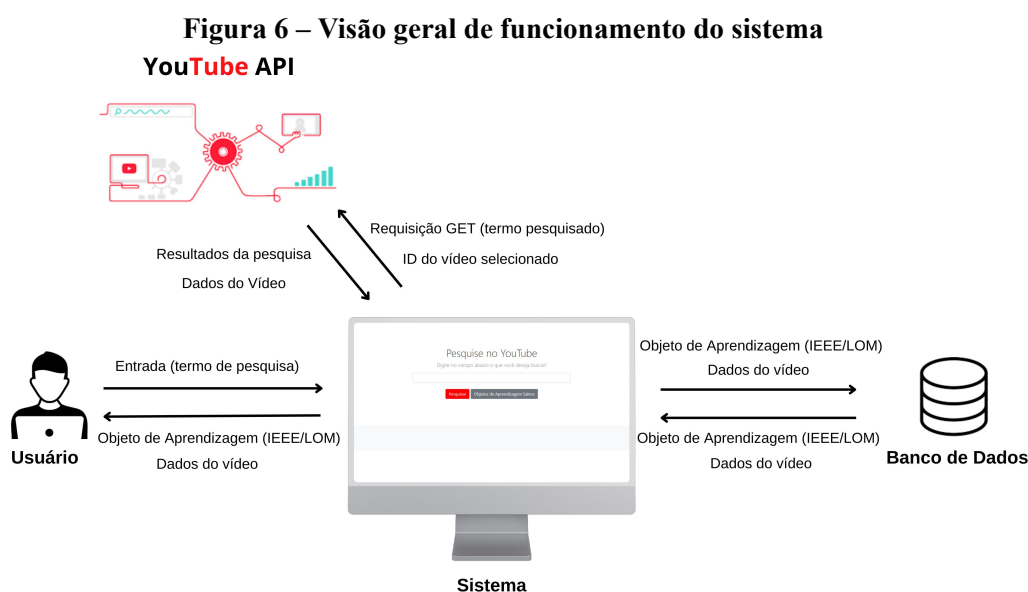
A interface do sistema foi toda baseada no *template engine* Jinja2 que é escrito com o próprio Python e que facilita a criação de páginas HTML para criar aplicações em Python. É dito *template engine* porque ele permite que linguagens de programação sejam incorporadas

⁶ O JSON está disponível em <<https://www.json.org/json-pt.html>>

em páginas HTML, o que possibilita que os desenvolvedores utilizem estruturas de condição, estruturas de repetição, herança e diversos outros recursos presentes apenas nas linguagens de programação nas páginas HTML. O Jinja2 foi de grande utilidade porque permitiu a fácil comunicação do *front-end* com o *back-end*.

5.3 Visão geral do Sistema

A Figura 6 demonstra a visão geral de funcionamento do sistema, apresentando o contato do usuário com o grupo. Demonstra ainda como é feita a comunicação com a API do YouTube, o que é salvo no banco de dados e o que o sistema apresenta ao usuário.



Fonte: Próprio autor

O funcionamento do sistema pode ser dividido em algumas etapas, em que:

1. O usuário acessa o sistema e realiza a pesquisa desejada;
2. O sistema envia a requisição feita pelo usuário para API do YouTube;
3. A API retorna a resposta com os resultados ao sistema e este mostra os resultados ao usuário;
4. O usuário pode escolher o Objeto de Aprendizagem do vídeo que vai ser salvo;
5. Ao fazer a escolha, o sistema envia a solicitação feita pelo usuário para API que retorna os dados ao sistema, dentre esses dados pode-se destacar a identificação do vídeo, o título do vídeo, a descrição, a categoria, o número de visualizações, o número de curtidas, entre outros dados;
6. Ao receber os dados da API o sistema abre a tela de Objetos de Aprendizagem Salvos e fornece opções ao usuário. As opções fornecidas permitem o usuário visualizar no YouTube o OA do vídeo que foi salvo, deletar o Objeto de Aprendizagem, editá-lo e visualizar o Objeto padronizado, no formato IEEE/LOM.

5.4 Funcionamento do Sistema

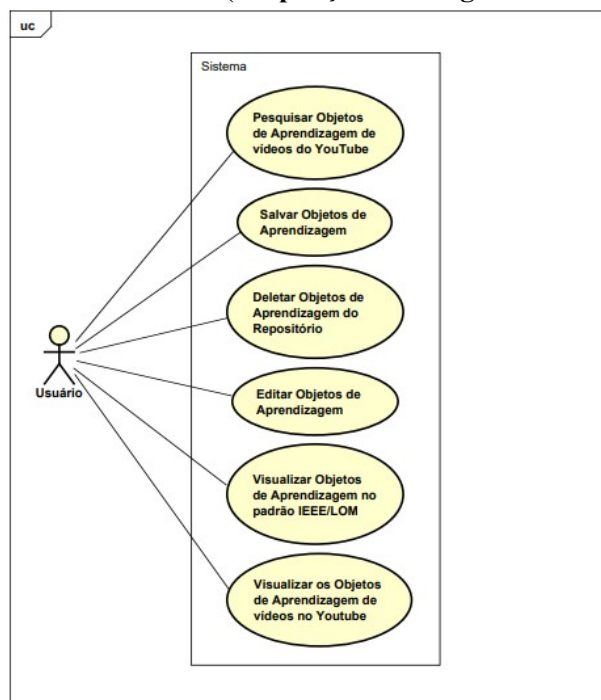
Para ficar mais explicativo como é o funcionamento do sistema, será apresentado um diagrama de casos de uso que é uma representação gráfica onde demonstra quais são as funções que o usuário pode realizar na aplicação. Também será apresentado um fluxograma que demonstra os processos existentes no sistema, quais as decisões que o usuário pode tomar; e quais devem ser as ações realizadas por meio destas decisões. Por fim, será mostrada as telas do sistema, simulando o seu funcionamento.

5.4.1 Caso de Uso do Sistema

A Figura 7 demonstra as ações realizadas pelo usuário no sistema. Dentre essas ações pode-se citar:

- Pesquisar Objetos de Aprendizagem de vídeos do Youtube;
- Salvar os Objetos de Aprendizagem no repositório;
- Deletar Objetos de Aprendizagem do repositório;
- Editar Objetos de Aprendizagem que foram salvos no repositório;
- Visualizar os Objetos de Aprendizagem organizados no padrão IEEE/LOM;
- Visualizar os Objetos de Aprendizagem de vídeos na plataforma do YouTube.

Figura 7 – Caso de Uso do Sistema (Ampliação da imagem encontra-se em Anexo A)



Fonte: Próprio autor

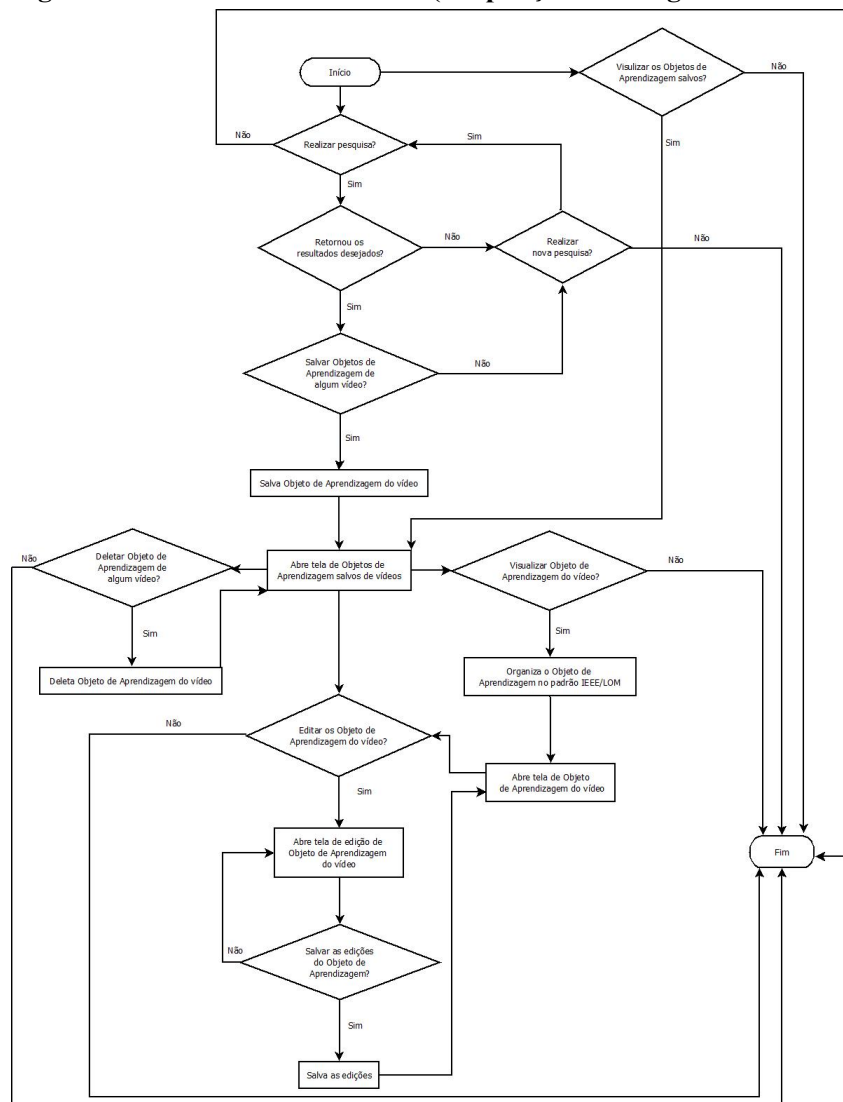
5.4.2 Fluxograma do Sistema

O fluxograma pode ser considerado um diagrama que descreve uma sequência de atividades e etapas que o sistema desempenha. Para exemplificar o funcionamento da aplicação foi utilizado o fluxograma de processos, que vai apresentar o passo a passo de tarefas realizadas pelo sistema. Vale ressaltar o significado representado por cada forma geométrica existente no fluxograma, em que:

- Retângulo arredondado - representa o início ou fim do fluxograma;
- Retângulo - representa um processo, considerada uma ação realizada no sistema;
- Losango - representa uma decisão que o usuário deve tomar no sistema;
- Linha - representa o fluxo, ou seja, a sequência que as etapas devem ocorrer.

A Figura 8 demonstra o fluxograma que representa os processos existentes no sistema:

Figura 8 – Fluxograma de Processos do Sistema (Ampliação da imagem encontra-se em Anexo B)



Fonte: Próprio autor

5.4.3 Telas do Sistema

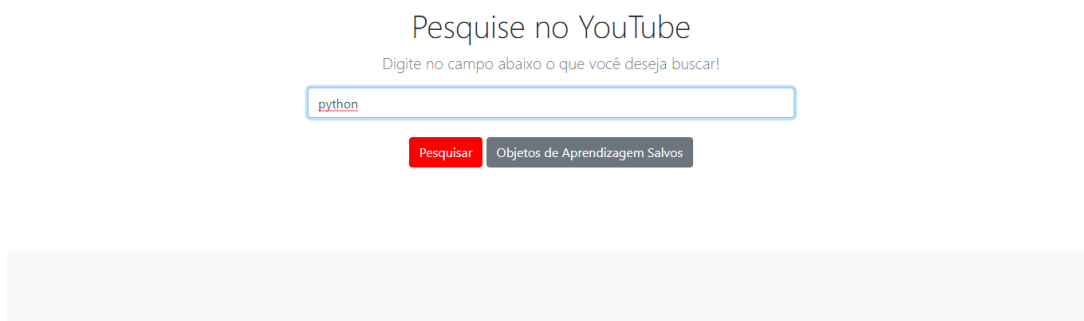
De início tem-se a primeira página da aplicação que é um mecanismo de pesquisa que possui dois botões no qual o botão de “Pesquisa” faz a busca do termo digitado pelo usuário, já o botão de “Objetos de Aprendizagem Salvos” redireciona o usuário para a página onde fica guardado os Objetos de Aprendizagem que foram salvos pelo usuário. Para começar é necessário o usuário realizar a digitação do termo que ele deseja e apertar o botão de Pesquisar. Na Figura 9 é mostrada a tela principal do sistema e na Figura 10 foi digitado o termo *Python* e clicado no botão Pesquisar:

Figura 9 – Tela Principal do Sistema



Fonte: Próprio autor

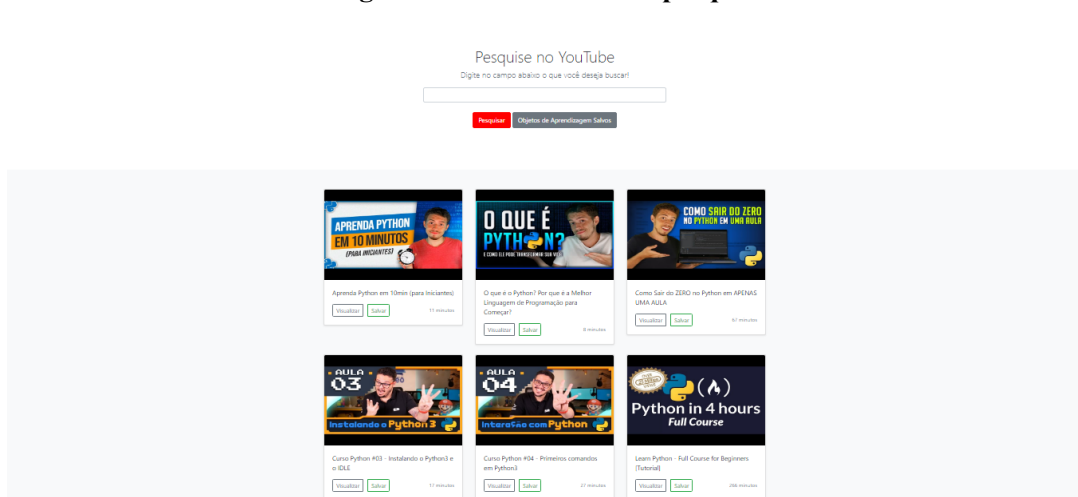
Figura 10 – Termo a ser pesquisado



Fonte: Próprio autor

A Figura 11 demonstra os resultados retornados pela API referentes ao termo que o usuário digitou:

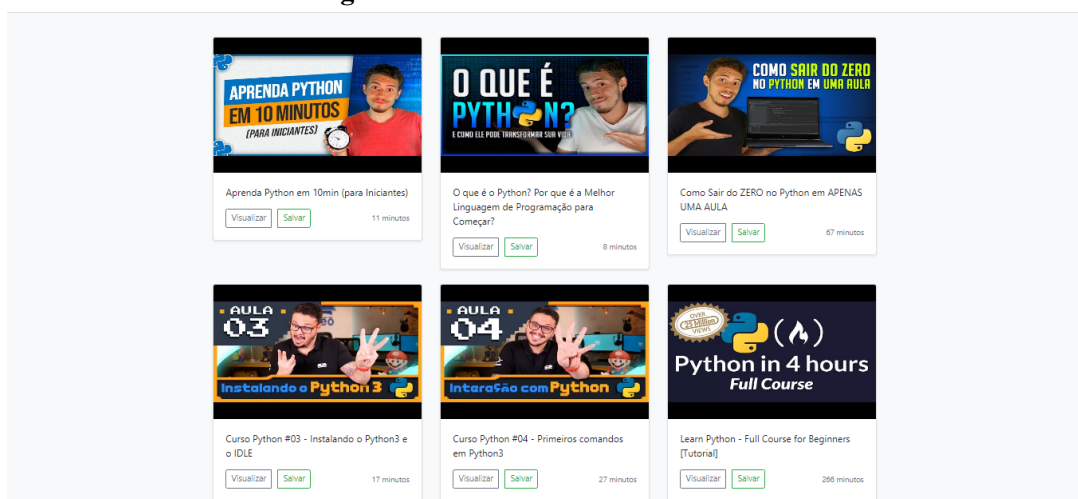
Figura 11 – Resultados da pesquisa



Fonte: Próprio autor

É possível perceber que em cada resultado há dois botões que são os denominados: “Visualizar” e “Salvar”. De forma sucinta, o botão de “Visualizar” faz o redirecionamento do usuário para a página do vídeo na plataforma do YouTube (para caso o mesmo deseje assistir ao vídeo), já o botão “Salvar” realiza o salvamento do Objeto de Aprendizagem relacionado ao vídeo e redireciona para a página de “Objetos de Aprendizagem Salvos”, como é demonstrado na Figura 12:

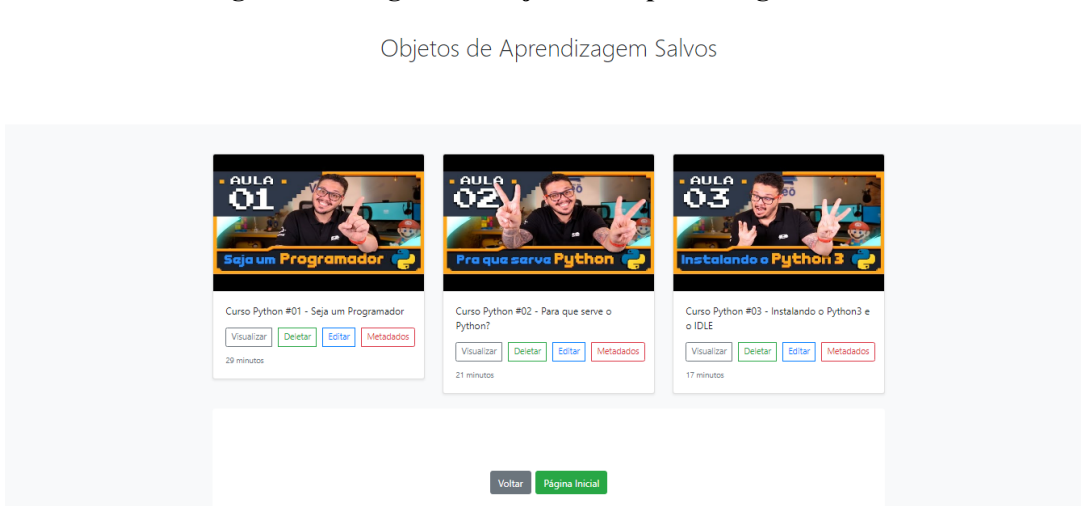
Figura 12 – Botões Visualizar e Salvar



Fonte: Próprio autor

Na Figura 13 é mostrada a página de “Objetos de Aprendizagem Salvos” e demonstra qual Objeto de Aprendizagem foi salvo pelo usuário, o ponto de atenção agora é referente aos botões existentes no Objeto de Aprendizagem. A seguir serão explicados o funcionamento de cada um:

Figura 13 – Página de Objetos de Aprendizagem Salvos



Fonte: Próprio autor

1- Visualizar: esse botão como mencionado anteriormente fará o redirecionamento do usuário para a página do vídeo na plataforma do YouTube, conforme Figura 14:

Figura 14 – Visualização do vídeo no YouTube

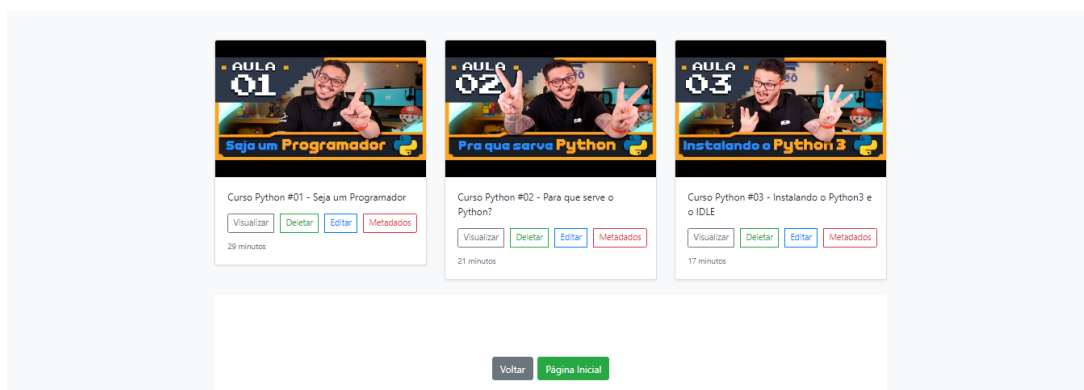


Fonte: Próprio autor

2- Deletar: este botão fará a remoção do Objeto de Aprendizagem salvo pelo usuário da página de “Objetos de Aprendizagem Salvos” além de estar removendo o mesmo do banco de dados, como mostra a Figura 15 e Figura 16:

Figura 15 – Antes da ação de deletar

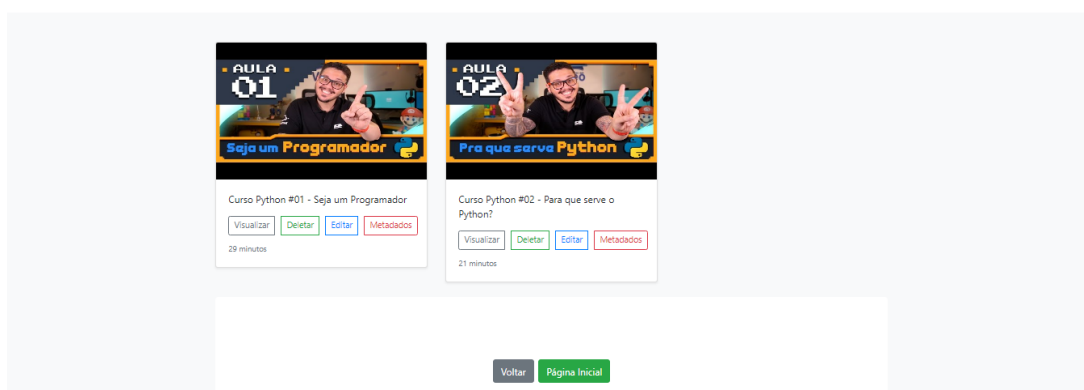
Objetos de Aprendizagem Salvos



Fonte: Próprio autor

Figura 16 – Depois da ação de deletar

Objetos de Aprendizagem Salvos



Fonte: Próprio autor

3- Editar: o botão “Editar” permite ao usuário fazer a edição dos metadados através das informações retornadas pela API e que organizam o Objeto de Aprendizagem, conforme Figura 17:

Figura 17 – Campos para edição

IEEE LOM:

```

"general":{
  "Identifier": "S9UPHppG6So",
  "Title": "Curso Python #01 - Seja um Programador",
  "Catalog_Entry":{
    "Catalogue": "E7",
    "Entry": "None"
  },
  "Language": "None",
  "Description": "Você está pensando em ser um Programador e não sabe por onde começar? Pois nessa aula vamos te dar todas as informações e mostrar o caminho a seguir. Gostou da aula? Então torne-se um Gafanhoto APOIADOR do cursoemvideo acessando o site cursoemvideo.com/apoie",
  "Keywords": "None",
  "Coverage": "None",
  "Structure": "Linear, Hierárquico",
  "Aggregation_Level": "3"
},
"Life_Cycle":{
  "Version": "None",
  "Status": "None",
  "Contribute":{
    "Role": "None",
    "Entity": "None"
  }
},
"Date": "2017-04-03T14:00:03Z"

```

Fonte: Próprio autor

O usuário poderá escolher entre salvar as alterações feitas nos metadados ou cancelar essas alterações através dos botões “Salvar” e “Cancelar” respectivamente, conforme Figura 18:

Figura 18 – Botões cancelar e salvar edição

```

"Kind": "None",
"Resource":{
  "Identifier": "None",
  "Description": "None",
  "Catalog_Entry": "None"
},
"Annotation":{
  "Person": "None",
  "Date": "2017-04-03T14:00:03Z",
  "Description": "None"
},
"Classification":{
  "Purpose": "None",
  "Taxon_Path":{
    "Source": "None",
    "Taxon":{
      "ID": "S9UPHppG6So",
      "Entry": "https://www.youtube.com/watch?v=S9UPHppG6So"
    }
  }
},
"Description": "None",
"Keywords": "None"
}

```

Cancelar Alteração Salvar Alteração

Fonte: Próprio autor

4- Metadados: o botão de “Metadados” faz a organização das informações dos Objetos de Aprendizagem retornadas pela API e as coloca no padrão IEEE/LOM, como mostrado na Figura 19:

Figura 19 – Metadados do vídeo

```

IEEE LOM:

"General":{
  "Identifier":"SSUPhpps6so,
  "Title":curso Python #01 - Seja um Programador,
  "Catalog_Entry":{
    "Catalogue": 27,
    "Entry": None
  },
  "Language": None,
  "Description":"Você está pensando em ser um Programador e não sabe por onde começar? Pois nessa aula vamos te dar t
Gostou da aula? Então torne-se um Gafanhoto APOIADOR do cursoemvídeo acessando o site cursoemvideo.com/apoie
Segundo um levantamento feito pela Code.org, em menos de 10 anos, serão necessários 1 milhão e 400 mil programadores no mercado de trabalho, mas
A programação é necessária em vários segmentos atualmente, seja em dispositivos computacionais tradicionais, em portáteis, eletrodomésticos inte
Um bom programador consegue resolver problemas de forma mais rápida, é mais criativo, tem espírito de comunidade e aprende as coisas mais rápido
E é uma grande MENTIRA dizer que para saber programar é preciso aprender Inglês ou ser um expert em Matemática. Isso tudo é uma LENDA!
O ramo de programação não escolhe idade, sexo, condição financeira ou área de especialização. Todos podem aprender a programar, até mesmo aquele
Neste vídeo você vai ver como é fácil começar a programar e ganhar estímulo para continuar na carreira.

Aula do Curso de Programação em Linguagem Python criado pelo professor Gustavo Guanabara para o portal Cursoemvideo.com.

Curso em vídeo
Seja um apoiador: http://cursoemvideo.com/apoie
Site: http://www.cursoemvideo.com
YouTube: http://www.youtube.com/cursoemvideo
Facebook: http://www.facebook.com/cursoemvideo
Twitter: http://twitter.com/cursoemvideo
Google+: http://plus.google.com/1126655837454979080,
"Keywords": None,

```

Fonte: Próprio autor

O usuário terá a opção de voltar para página de “Objetos de Aprendizagem Salvos” através do botão “Voltar” ou realizar a edição dos metadados através do botão “Editar”, conforme Figura 20:

Figura 20 – Botões Voltar e Editar

```

"Description": None,
"Language": None
},
"Rights":{
  "Cost": None,
  "Copyright_and_other_restrictions": True,
  "Description": None,
},
"Relation":{
  "Kind": None,
  "Resource":{
    "Identifier": None,
    "Description": None,
    "Catalog_Entry": None
  }
},
"Annotation":{
  "Person": None,
  "Date": 2017-04-03T14:00:03Z,
  "Description": None
},
"Classification":{
  "Purpose": None,
  "Taxon_path":{
    "source": None,
    "taxon":{
      "ID": SSUPhpps6so,
      "Entry": https://www.youtube.com/watch?v=SSUPhpps6so
    }
  }
},
"Description": None,
"Keywords": None
}

```

Voltar
Editar

Fonte: Próprio autor

6 RESULTADOS ALCANÇADOS

Pode-se destacar como resultado alcançado, além do sistema, os Objetos de Aprendizagem organizados no padrão IEEE/LOM, sendo este o objetivo principal pela qual o sistema foi desenvolvido. Nesta seção é demonstrado o resultado do teste realizado no sistema que retornou o Objeto de Aprendizagem e sua organização no padrão IEEE/LOM. A Figura 21 apresenta como é estruturado o padrão IEEE/LOM e por conseguinte é demonstrado a correspondência de cada metadado do Objeto de Aprendizagem extraído do vídeo com a respectiva categoria que melhor o descreve:

Figura 21 – Categorias de Metadados IEEE LOM

Categoria	Descrição
Geral	Apresenta as informações que descrevem o objeto de aprendizagem como um todo: dados de identificação (catálogo e entrada), título, idioma, descrição, palavras-chave, cobertura, estrutura, nível de agregação.
Ciclo de Vida	Descreve as características relacionadas com a história e o estado atual dos objetos de aprendizagem e como ele foi afetado durante sua evolução, tais como: versão, <i>status</i> , contribuintes (papel, entidade, data).
Meta-Metadados	Agrupa informações sobre a própria instância do metadado: identificador (catálogo, entrada), contribuintes (papel, entidade e data), esquema de metadados e idioma.
Técnico	Descreve as exigências e características técnicas do objeto educacional, ou seja, os dados técnicos, tais como: formato, tamanho, localização, requisitos de sistema operacional, comentários sobre instalação, requisitos para outras plataformas, duração.
Educacional	Apresenta as características educacionais e pedagógicas do objeto tipo de interatividade, tipo de recurso de aprendizagem, nível de interatividade, densidade semântica, público alvo, contexto, faixa etária recomendada, dificuldade, tempo de aprendizagem típica, descrição e idioma.
Direitos Autorais	Agrupa os direitos de propriedade intelectual e condições de uso para o objeto, tais como: custos, direitos autorais e outras restrições e descrição.
Relação	Descreve características que relacionam o objeto de aprendizagem com outros objetos educacionais correlacionados, tais como: tipo e recurso (identificador – catálogo e entrada e descrição).
Anotação	Provê comentários do uso educacional do objeto e informações sobre quando e por quem foram criados os comentários. Essa categoria é composta por: entidade, data e descrição.
Classificação	Determina em que local o recurso será colocado, dentro de um sistema de classificação específico, por meio de: propósito, caminho da taxonomia (identificador e entrada), descrição e palavra-chave.

6.1 Objeto de Aprendizagem gerado pelo sistema

O teste realizado no sistema propôs fazer a pesquisa do termo “*Python*” na qual obteve-se diversos resultados provenientes da API. Para tanto, foi escolhido o resultado de título “Curso Python 01 - Seja um Programador” salvando o mesmo no repositório de Objetos de Aprendizagem. Logo após foi clicado no botão “metadados” que organizou o Objeto de Aprendizagem extraído do vídeo no padrão IEEE/LOM, com isso o seguinte resultado foi obtido:

```
"General": {
  "Identifier": S9uPNppGsGo,
  "Title": Curso Python #01 - Seja um Programador,
  "Catalog_Entry": {
    "Catalogue": 27,
    "Entry": None
  },
  "Language": None,
  "Description": Curso em Vídeo
Seja um apoiador: http: //cursoemvideo.com/apoie
Site: http: //www.cursoemvideo.com
YouTube: http: //www.youtube.com/cursoemvideo
Facebook: http: //www.facebook.com/cursosemvideo
Twitter: http: //twitter.com/cursosemvideo
Google+: http: //plus.google.com/112666558837414979080,
  "Keywords": None,
  "Coverage": None,
  "Structure": Linear, Hierárquico,
  "Aggregation_Level": 3
},
[...]
```

O resultado completo do Objeto de Aprendizagem encontra-se no Anexo C.

Com o resultado obtido pode-se dizer que o objetivo principal do sistema foi alcançado com êxito, pois a partir de um vídeo do YouTube foi feita a organização de um Objeto de Aprendizagem estruturado no padrão IEEE/LOM e assim fazendo a gravação do mesmo no repositório.

7 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar uma solução que utiliza vídeos do YouTube como fonte para a organização de Objetos de Aprendizagem seguindo o padrão IEEE/LOM, com a finalidade de proporcionar ao usuário informações estruturadas e organizadas, facilitando seu processo de aprendizagem. Dessa forma, o usuário terá uma noção mais precisa sobre se o vídeo escolhido atenderá às suas expectativas e objetivos de aprendizagem.

É importante ressaltar que a linguagem *Python* e o *Micro-Framework Flask* foram ferramentas de grande valia para que o propósito final do sistema fosse alcançado. A criação da aplicabilidade ocorreu sem muitas complicações com o uso das ferramentas e ainda foi permitido alcançar um resultado satisfatório através de um sistema construído para ser simples e intuitivo.

O sistema desenvolvido não foi colocado em um ambiente de produção para que diferentes usuários fizessem o uso do mesmo, porém foram realizados testes que garantem um bom funcionamento sem grandes problemas, visto que por ter sido construído com o intuito de ser simples ele não exige muitos recursos para seu funcionamento, o que pode vir a torná-lo escalável.

O presente trabalho proporcionou um conhecimento aprofundado no desenvolvimento *Web* no que tange a linguagem *Python*, requisições a API e manipulação de dados. Além disso, foi possível conhecer mais dos Objetos de Aprendizagem e sua importância para o estudo.

Como sugestões futuras ao sistema, pretende-se criar uma forma de poder assistir ao vídeo na própria aplicação e implementar um controle de usuário para que cada um possa criar seus próprios repositórios. Essa criação traz consigo a finalidade de organizar melhor seus objetivos de aprendizado, além de poder compartilhar seus repositórios com outros usuários.

REFERÊNCIAS

- ALVES, R. C. V. *et al.* Metadados como elementos do processo de catalogação. Universidade Estadual Paulista (UNESP), 2010.
- ANTUNES, D. **DESENVOLVIMENTO BACK-END: UM GUIA PARA EMPREENDEDORES**. 2022. Disponível em: <<https://www.gobacklog.com/blog/back-end-guia-para-empresendedores/>>. Acesso em: 01 agosto 2022.
- BARRO, B. B. **O que são Frameworks**. 2022. Disponível em: <<https://www.hostinger.com.br/tutoriais/frameworks>>. Acesso em: 03 abril 2022.
- BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. [S.l.]: Novatec Editora, 2014.
- BRITO, R. W. Bancos de dados nosql x sgbd's relacionais: análise comparativa. **Faculdade Farias Brito e Universidade de Fortaleza**, 2010.
- CARDOSO, R. Bases de dados nosql, instituto superior de engenharia do porto. **Outubro**, 2012.
- CORBIÈRE, A.; CHOQUET, C. Designer integration in training cycles: Ieee ltsa model adaptation. In: **Proceedings of the International Conference on Computer Aided Learning in Engineering Education (CALIE 2004), Grenoble, France**. [S.l.: s.n.], 2004.
- HARMAN, K.; KOOHANG, A. **Learning objects: standards, metadata, repositories, and LCMS**. [S.l.]: Informing Science, 2007.
- HASHTAG. **BIBLIOTECAS DO PYTHON: CONHEÇA AS MELHORES POR FINALIDADE!** 2022. Disponível em: <<https://www.hashtagtreinamentos.com/bibliotecas-mais-importantes-do-python>>. Acesso em: 03 abril 2022.
- JONES, E. R. Implications of scorm and emerging e-learning standards on engineering education. In: **2002 GSW**. [S.l.: s.n.], 2022.
- LATORRE, M. **Historia de las Web, 1.0, 2.0, 3.0 y 4.0**. Universidad Marcelino Champagnat. 2018.
- LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, sn, v. 10, n. 1, p. 11, 2011.
- MONTORO, A. F. **Python 3 al discovered**. [S.l.: s.n.], 2012.
- PINHEIRO, C. Ângelo. **O que é um repositório de objeto de aprendizagem?** 2022. Disponível em: <<https://todasasrespostas.pt/o-que-e-um-repositorio-de-objeto-de-aprendizagem>>. Acesso em: 01 agosto 2022.
- PINTAR, P. H. R. Um estudo exploratório acerca de banco de dados nosql comparado aos bancos de dados relacionais. **FÓRUM CIENTÍFICO DA FEMA-ANAIS**, p. 15, 2016.
- PÖTTKER, L. M. V.; FERNEDA, E.; MOREIRO-GONZÁLEZ, J. A. Mapeamento relacional entre padrões de metadados educacionais. **Perspectivas em Ciência da Informação**, SciELO Brasil, v. 23, p. 25–38, 2018.

RILEY, J. Understanding metadata. **Washington DC, United States: National Information Standards Organization** (<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>), v. 23, 2017.

SILVA, E. L. d.; CAFÉ, L.; CATAPAN, A. H. Os objetos educacionais, os metadados e os repositórios na sociedade da informação. **Ciência da Informação**, SciELO Brasil, v. 39, p. 93–104, 2010.

SILVA, R. S. da. *Objetos de aprendizagem para educação a distância*. 2011.

SIMÃO, J. P. S.; PEREIRA, J.; ALVES, J. B. da M.; SILVA, J. B. da. Padrões para interoperabilidade de dados em aplicações educacionais. **Anais do Simpósio Ibero-Americano de Tecnologias Educacionais**, p. 216–222, 2018.

SOUZA¹, D. d. S. R.; ALBUQUERQUE, B. C. D. de; SILVA, K. K. de A. *Repositórios de objetos de aprendizagem: Uma estratégia colaborativa*. 2015.

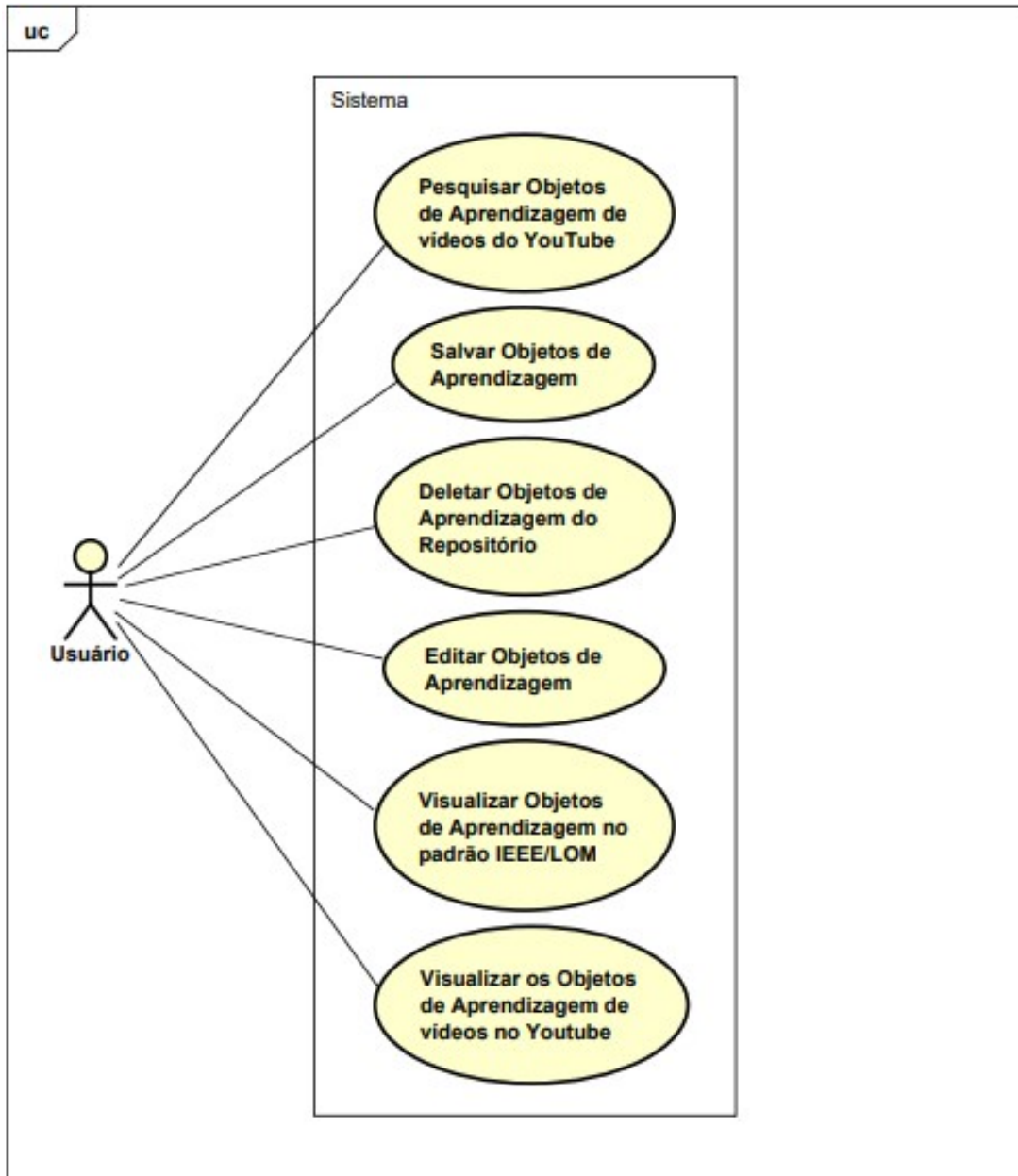
TAROUCO, L. M. R.; COSTA, V. M. d.; AVILA, B. G.; BEZ, M. R.; SANTOS, E. F. d. *Objetos de aprendizagem: teoria e prática*. Evangraf, 2014.

WILEY, D. A. *et al.* Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. **The instructional use of learning objects**, v. 2830, n. 435, p. 1–35, 2000.

YOUTUBE. **API Reference**. 2019. Disponível em: <<https://developers.google.com/youtube/v3/docs?hl=pt-br>>. Acesso em: 03 abril 2022.

ANEXO A – FIGURA 7 AMPLIADA

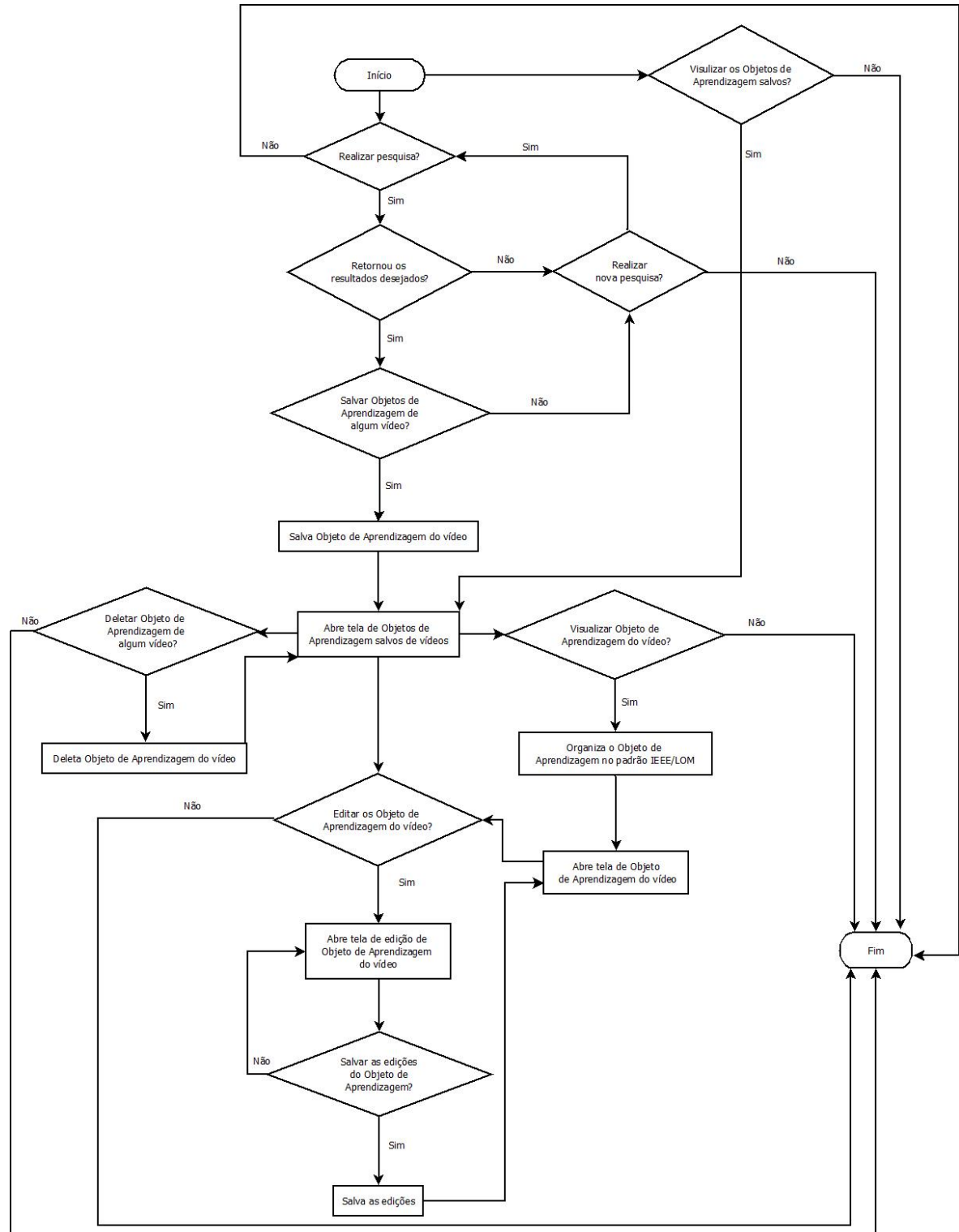
Figura 22 – Caso de Uso do Sistema Ampliado



Fonte: Próprio autor

ANEXO B – FIGURA 8 AMPLIADA

Figura 23 – Fluxograma de Processos do Sistema Ampliado



Fonte: Próprio autor

ANEXO C – OBJETO DE APRENDIZAGEM GERADO PELO SISTEMA

```

"General": {
  "Identifier": S9uPNppGsGo,
  "Title": Curso Python #01 - Seja um Programador,
  "Catalog_Entry": {
    "Catalogue": 27,
    "Entry": None
  },
  "Language": None,
  "Description": Curso em Vídeo
  Seja um apoiador: http: //cursoemvideo.com/apoie
  Site: http: //www.cursoemvideo.com
  YouTube: http: //www.youtube.com/cursoemvideo
  Facebook: http: //www.facebook.com/cursoemvideo
  Twitter: http: //twitter.com/cursoemvideo
  Google+: http: //plus.google.com/112666558837414979080,
  "Keywords": None,
  "Coverage": None,
  "Structure": Linear, Hierárquico,
  "Aggregation_Level": 3
},
"Life_Cycle": {
  "Version": None,
  "Status": None,
  "Contribute": {
    "Role": None,
    "Entity": None,
    "Date": 2017-04-03T14: 00: 03Z
  }
},
"Meta_metadata": {
  "Identifier": S9uPNppGsGo,
  "Catalog": {
    "Catalog": 27,
    "Entry": None
  },
  "Contribute": {
    "Role": Curso em Vídeo,

```

```
        "Entity": YouTube,
        "Date": 2017-04-03T14: 00: 03Z
    },
    "Metadata_Scheme": IEEE LOM,
    "Language": None
},
"Technical": {
    "Format": None,
    "Size": None,
    "Location": None,
    "Requirements": {
        "Type": None,
        "Name": None,
        "Min_version": None,
        "Max_version": None
    },
    "Installation_Remarks": None,
    "Other_platform_requirements": None,
    "Duration": 29 minutos
},
"Educational": {
    "Interactivity_Type": Expositivo,
    "Learning_Resource_Type": Vídeo,
    "Interactivity_Level": Baixo,
    "Semantic_Density": Médio,
    "Intended_end_user_role": Aprendiz,
    "Context": None,
    "Typical_Age_Range": None,
    "Difficulty": None,
    "Typical_learning_Time": None,
    "Description": None,
    "Language": None
},
"Rights": {
    "Cost": None,
    "Copyright_and_other_restrictions": True,
    "Description": None,
},
"Relation": {
```

```
"Kind": None,
"Resource": {
  "Identifier": None,
  "Description": None,
  "Catalog_entry": None
},
"Annotation": {
  "Person": None,
  "Date": 2017-04-03T14: 00: 03Z,
  "Description": None
},
"Classification": {
  "Purpose": None,
  "Taxon_path": {
    "Source": None,
    "Taxon": {
      "ID": S9uPNppGsGo,
      "Entry": https://www.youtube.com/watch?v=S9uPNppGsGo
    }
  }
},
"Description": None,
"Keywords": None
}
```