

Etec de Hortolândia

Terceiro ano do curso Técnico de Desenvolvimento de Sistemas

Paulo Eduardo Ferreira Junior

Vicente Matheus Collin Pedroso

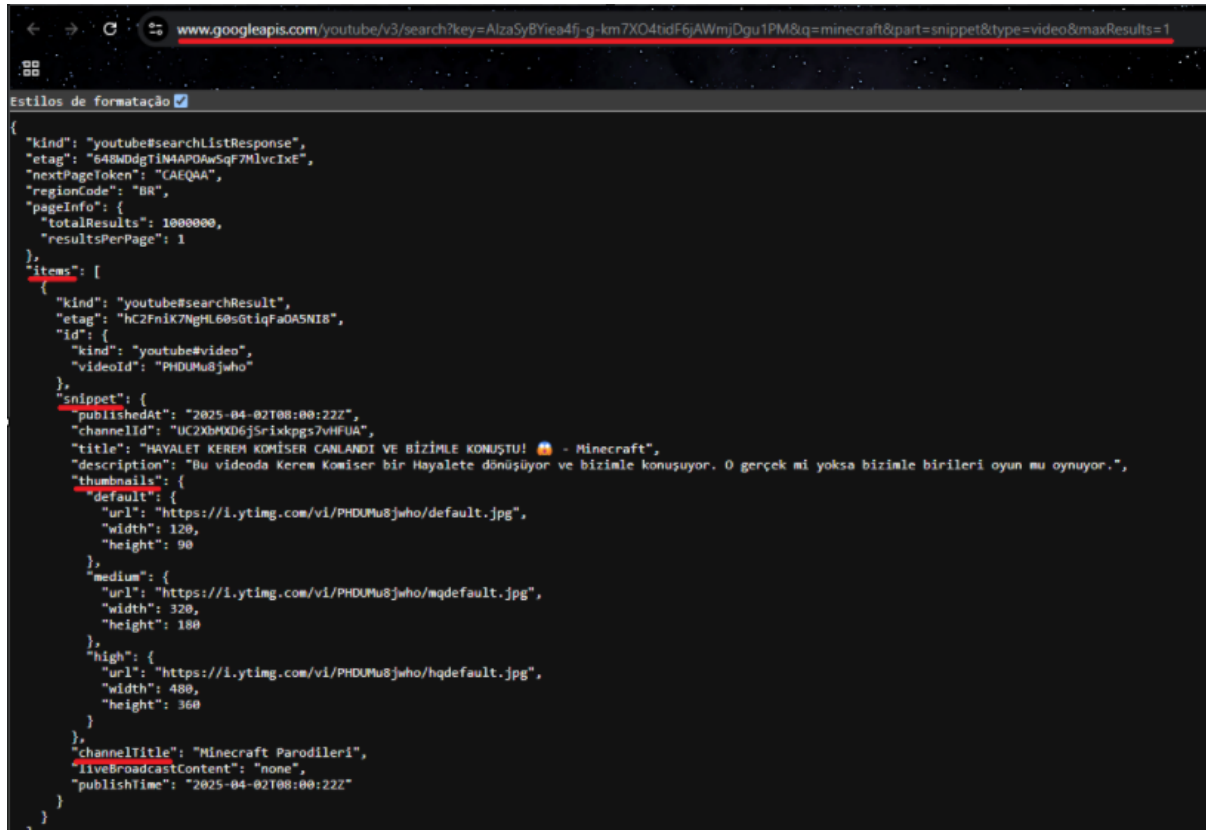
iWatch

Aplicação com consumo de APIs para entendimento básico de retornos

**Hortolândia
2025**

DESCRIÇÃO DA APLICAÇÃO

Aplicação desenvolvida no projeto iWatch consome uma API do Google que retorna informações e dados correspondentes a vídeos do YouTube por meio dos retornos fornecidos em JSON, conforme o exemplo da imagem abaixo:

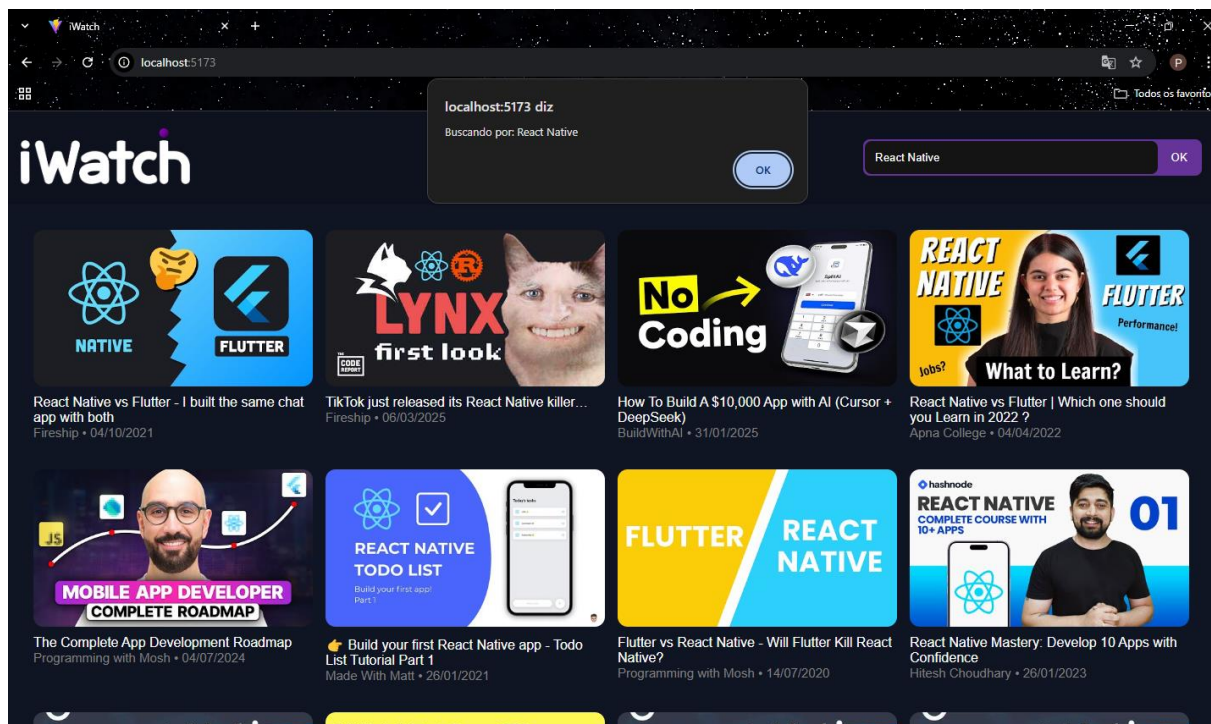


O YouTube Data API (API utilizada) é uma API muito abrangente e que fornece informações para diversos segmentos de aplicação. Na proposta desenvolvida pela dupla, os elementos selecionados para integrar o projeto corresponderam ao retorno da parte visual, como as thumbnails (capa dos vídeos), título, nome do canal e data de publicação. O objetivo do software é ser um catálogo de vídeos onde é possível pesquisar temas por palavras-chave em uma barra de pesquisa, que retorna vídeos de quaisquer nichos filtrados por ordem de relevância (visualizações).

PRINTS DAS TELAS

```
VideosLista.jsx X
src > componentes > VideosLista.jsx > ...
1  import VideoCard from './VideoCard'
2  import { useEffect, useState } from 'react'
3  import estilos from './VideosLista.module.css'
4
5  function VideoLista({ searchQuery }) {
6    const [videosAPI, setVideosAPI] = useState([]) // Retorno da API
7    const [erro, setErro] = useState('') // Mensagem de Erro
8
9    const chaveYT = 'AIzaSyBYiea4fj-g-km7X04tidF6jAWmjDgu1PM'
10   const buscaPadrao = '' // Busca padrão ao entrar na page
11
12   const buscaVideos = async (termoBusca) => {
13
14     try {
15       const respostaJSON = await fetch(
16         `https://www.googleapis.com/youtube/v3/search?key=${chaveYT}&q=${termoBusca}&part=snippet&order=viewCount`
17       )
18       const respostaJS = await respostaJSON.json()
19
20       if (respostaJS.items) {
21         setVideosAPI(respostaJS.items)
22         console.log(respostaJS)
23       } else {
24         setErro("Nenhum vídeo encontrado.")
25       }
26     } catch (erro) {
27       setErro(`A busca por vídeos falhou! (Erro: ${erro.message})`)
28     }
29   }
30
31
32
33
34   useEffect(() => {
35     const termoBusca = searchQuery || buscaPadrao
36     buscaVideos(termoBusca)
37   }, [searchQuery])
38
39   return (
40     <main className={estilos.videosContainer}>
41       {erro && <p style={{ color: 'red' }}>{erro}</p>}
42       {videosAPI.map(cadaVideo => (
43         <VideoCard key={cadaVideo.id.videoId} propsVideo={cadaVideo} />
44       ))}
45     </main>
46   )
47 }
48
49 export default VideoLista
50
```

Capturas de tela do componente VideosLista.jsx



Captura de tela do projeto executando no navegador
(e retornando os elementos supracitados).

LINKS DE DOCUMENTAÇÃO DA API

- Solicitação HTTP: GET <https://www.googleapis.com/youtube/v3/videos>
- URL da API (com chave e filtragem de três resultados para amostragem):
<https://www.googleapis.com/youtube/v3/search?key=AlzaSyBYiea4fj-g-km7XO4tidF6jAWmjDgu1PM&part=snippet&type=video&maxResults=3>
- Sobre a API (e componentes em geral):
<https://developers.google.com/youtube/v3/docs?apix=true&hl=pt-br>
- Parte sobre a inserção de informações provenientes de vídeos:
<https://developers.google.com/youtube/v3/docs/videos/list?hl=pt-br>