

Experimento de Processamento de Músicas com Python Notebook e Aplicativo IoT

Felipe Almeida e Victor Hayashi

Resumo—Proporcionar um aprendizado ativo em processamento de sinais é um desafio educacional relevante. Recursos Educacionais Abertos (REA) podem fomentar atividades práticas que auxiliam o aprendizado da teoria. Neste artigo, um experimento de processamento de músicas em um Python Notebook para execução local ou na nuvem da Google é apresentado para complementar a educação teórica em processamento de sinais. O experimento de código aberto está conectado a uma plataforma IoT em nuvem para fomentar integrações externas em projetos dos alunos, e as atividades são organizadas de acordo com Objetivos de Aprendizagem conforme a Taxonomia de Bloom.

Palavras-Chave—Processamento de Sinais de Áudio, Educação, IoT, Cloud.

Abstract—Providing active learning in signal processing is a relevant educational challenge. Open Educational Resources (OER) might foster practical activities that assist in student learning. In this article, a music processing experiment in a Python Notebook with local and cloud execution alternatives is presented to complement the theoretical education in signal processing. The open-source experiment is also connected to a cloud IoT platform to foster external integrations in students' projects, and activities are organized according to Learning Objectives according to Bloom's Taxonomy.

Keywords—Audio Signal Processing, Education, IoT, Cloud.

I. INTRODUÇÃO

O processo de ensino-aprendizagem em processamento de sinais para alunos de graduação requer experimentos práticos. Além de motivar, a conexão com a realidade por meio de aplicações usuais presentes no cotidiano dos estudantes pode ajudar os discentes a obter intuições importantes para o entendimento de conceitos apresentados em aulas expositivas [1].

Neste contexto, este artigo descreve um experimento de processamento de músicas para facilitar a conexão com a realidade para os estudantes.

O texto está organizado da seguinte forma: aspectos pedagógicos e teóricos são abordados na Seção II, enquanto a Seção III apresenta os materiais e métodos empregados. A Seção IV contém a descrição da solução proposta. Os resultados da implementação em ambiente local e na nuvem da Google estão apresentados na Seção V, e uma discussão em relação aos trabalhos relacionados é apresentada na Seção VI. A Seção VII conclui o trabalho, apresentando as contribuições e perspectivas de trabalhos futuros.

Felipe Almeida e Victor Hayashi, Departamento de Computação e Sistemas Digitais (PCS), Escola Politécnica da Universidade de São Paulo, São Paulo-SP, e-mails: {felipe.valencia.almeida, victor.hayashi}@usp.br

II. EMBASAMENTO TEÓRICO

Aprendizado ativo pode ser entendido como qualquer atividade de aprendizagem realizada por alunos que não seja a tradicional forma passiva de assistir a aulas de um professor. As atividades relacionadas a um aprendizado ativo podem ser realizadas individualmente ou em grupo, e podem fomentar a aplicação de materiais do curso a novos problemas ou em situações do “mundo real” [7].

Recursos Educacionais Abertos (REA) são materiais disponíveis em formato digital de forma livre para uso e reuso por estudantes e educadores em aulas e pesquisa. Podem ser conteúdos educacionais como módulos e materiais de aula, recursos de implementação como licenças que permitem a publicação aberta dos materiais e ferramentas [11].

A ideia principal da Taxonomia de Bloom é que educadores desejam que os estudantes atinjam determinados Objetivos de Aprendizagem após o processo de aprendizagem. Estes Objetivos de Aprendizagem podem ser organizados em uma taxonomia, do mais simples até o mais complexo. Os níveis cognitivos da taxonomia são: conhecimento, compreensão, aplicação, análise, síntese, criação e avaliação, sendo este último relacionado ao pensamento crítico. Cada objetivo está associado a um verbo, como por exemplo o verbo “listar” no nível de conhecimento e o verbo “criticar” no nível de avaliação. É esperado que cada Objetivo de Aprendizagem esteja também relacionado a um comportamento observável do aluno [10].

Project Based Learning (PBL) é um método para aprendizado ativo centrado em estudantes para fomentar autonomia, colaboração e reflexão considerando práticas do “mundo real”. É baseado nos princípios de que o aprendizado é específico ao contexto; de que alunos estão envolvidos de forma ativa no processo de aprendizado; e que alunos atingem seus objetivos por meio de interações sociais e colaboração [13].

O processamento de áudio pode ser realizado pela análise de parâmetros *chroma* [17], que permitem a análise de músicas a partir da distribuição de seus semi-tons, sendo utilizados para sincronização e análise da estrutura destes tipos de áudio [16]. Sua representação visual, denominada *chromagrama*, é uma variação das distribuições usuais para percepção de tons, que apresenta a distribuição da energia espectral em cada um dos 12 semi-tons da oitava [2]. Considerando o conjunto dos 12 *chromas* relacionados aos tons $\{C, C\#, D, \dots, B\}$, um parâmetro *chroma* [16] pode ser representado como um vetor de 12 dimensões da equação (1), onde $x(1)$ corresponde ao *chroma* C , $x(2)$ ao *chroma* $C\#$, $x(3)$ ao *chroma* D , e assim sucessivamente até $x(12)$, associado ao *chroma* B .

$$x = (x(1), x(2), \dots, x(12)) \quad (1)$$

Uma das possíveis definições de Internet das Coisas (IoT, *Internet of Things*) é uma rede de objetos inteligentes que possuem capacidade de auto-organização, compartilhamento de informações, dados e recursos, que podem reagir e atuar no caso de determinadas situações e mudanças em seu ambiente [15]. Neste trabalho, uma oportunidade explorada é a integração do experimento desenvolvido com uma plataforma de IoT em nuvem para suportar o compartilhamento das informações adquiridas, e tornar possível o uso desta informação em um projeto que pode ser desenvolvido pelo aluno em conjunto com seus colegas, considerando a aplicação do método PBL descrito anteriormente no curso.

III. MATERIAIS E MÉTODOS

Esta seção descreve os recursos computacionais utilizados neste trabalho. A Figura 1 apresenta uma visão da arquitetura destacando os recursos utilizados.

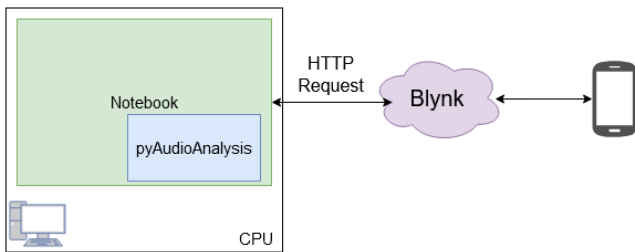


Fig. 1. Visão da arquitetura com ênfase nos recursos utilizados

A *pyAudioAnalysis* é uma biblioteca Python *open source* que provê uma gama de ferramentas para análise de áudio, como classificação, extração de parâmetros e segmentação [8]. Em relação à extração de parâmetros, que é o recurso utilizado neste trabalho, sua função *feature_extraction()* recebe como entradas uma amostra de áudio, a taxa de amostragem, o tamanho da janela deslizante e o passo da janela. Seu retorno é um array *numpy*, contendo 32 estruturas correspondentes a parâmetros distintos. É importante destacar as estruturas relacionadas ao *Chroma Vector*, que serão analisadas nos casos de teste. Informações sobre as demais estruturas podem ser obtidas em [8].

Notebooks são ambientes virtuais utilizados para computação interativa. Sua estrutura baseia-se em células, que podem ser preenchidas tanto com recursos textuais e visuais quanto com blocos de código escritos com linguagens de programação, como por exemplo a linguagem Python, permitindo o uso da biblioteca *pyAudioAnalysis*. Diversos produtos são disponíveis atualmente utilizando o conceito de notebooks, sendo os mais conhecidos do tipo *open-source*, como por exemplo o Jupyter Notebook [12] ou então o Google Colab [9]. Observa-se na literatura diversas tentativas de ensino via uso de notebooks [18] [20], devido ao caráter prático e os recursos diversos propiciados por esta aplicação. No caso do Google Colab por exemplo, pode-se destacar a vantagem de utilizar um ambiente externo, sem a necessidade de instalar qualquer recurso do computador local do aluno.

Blynk [3] é uma plataforma voltada para projetos de IoT. Seu desenvolvimento foi motivado por uma necessidade de uma plataforma simples, capaz de facilitar a integração entre múltiplos componentes, podendo ser sensores, microcontroladores, como o Arduino, computadores de pequeno porte, como o caso do Raspberry Pi, dentre outros. O uso de uma solução deste tipo reduz a complexidade de um projeto de IoT, pois é possível padronizar a comunicação entre os componentes e eliminar a necessidade de configuração de múltiplos protocolos de comunicação. A interface básica do Blynk é utilizada por meio do seu aplicativo, que pode ser obtido na Google Play (para dispositivos Android) ou na App Store (para dispositivos iOS).

É importante destacar a definição de *pino virtual*: ao contrário de um pino digital ou analógico, um pino virtual é uma abstração do Blynk que não possui equivalente físico. Ele corresponde a um endereço na nuvem do Blynk, onde os dados podem ser lidos ou escritos, promovendo assim uma maior flexibilidade na aplicação. Estes pinos podem ser lidos ou ter seu valor alterado facilmente por meio de requisições do tipo HTTP (*HTTP Requests*). Estas atuam como a forma de integrar os notebooks com o Blynk.

Requisições HTTP utilizam-se do protocolo de mesmo nome para realizar a comunicação entre clientes e servidores [14], atuando como a base da camada de aplicação da Internet. No contexto deste trabalho estas requisições são utilizadas para comunicar os dispositivos com a nuvem do Blynk. Os métodos utilizados podem ser do tipo GET, POST, PUT, HEAD, DELETE, PATCH e OPTIONS, sendo os dois primeiros os mais utilizados. Requisições do tipo GET geralmente são utilizadas para solicitar algum tipo de dados, podendo também no caso do Blynk serem utilizadas para alterar o valor de algum pino virtual. Desta forma, neste trabalho foi necessário apenas o uso das requisições do tipo GET, alterando a URL conforme o tipo de operação, podendo esta ser de leitura ou escrita.

IV. SOLUÇÃO PROPOSTA

A união dos recursos apresentados na Seção III segundo a visão apresentada na Figura 1 resultam em uma ferramenta didática que pode ser utilizada para promover um experimento de processamento de músicas conforme Objetivos de Aprendizagem propostos segundo a Taxonomia de Bloom. A ferramenta consiste em um notebook que utiliza-se da biblioteca *pyAudioAnalysis* para processar o áudio e realiza conexão com a nuvem do Blynk por meio de requisições HTTP. Este notebook pode ser executado tanto no Google Colab, por meio de um ambiente virtual alocado nos servidores da Google, quanto localmente, utilizando o Jupyter Notebook. Seu código é aberto para que possa ser reutilizado REA. O notebook permite uma participação ativa do aluno no processo de aprendizagem, pois este pode alterar seu conteúdo conforme suas necessidades e dúvidas. Por exemplo: se o curso seguir o método de PBL, após a aula o aluno pode utilizar a ferramenta em um projeto, e comprovar a utilidade da ferramenta apresentada em aula.

A Figura 2 apresenta o contexto de uma aula utilizando a ferramenta proposta. Em um primeiro momento (1) o professor

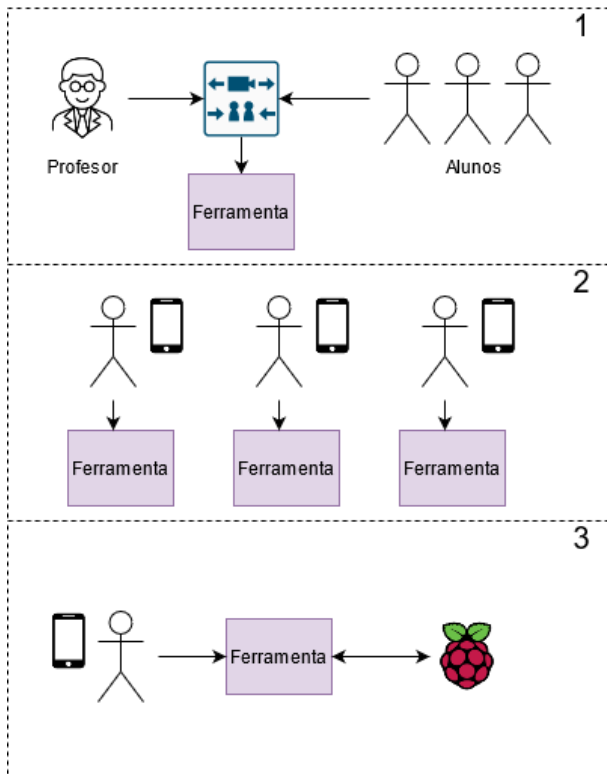


Fig. 2. Aula utilizando a ferramenta

apresenta para os alunos durante a aula a ferramenta no Google Colab, onde os alunos podem acompanhá-lo. Conceitos relacionados aos recursos computacionais utilizados e a teoria de processamento de áudio devem ser explicados e eventualmente reforçados, visando um melhor aproveitamento e autonomia por parte dos alunos. O professor pode neste momento utilizar uma amostra de áudio como teste, deixando para os alunos a tarefa de executar a ferramenta com outras amostras de áudio.

Em seguida (2) os alunos executam por conta própria a ferramenta, com outras amostras de áudio. Esta execução pode ser tanto pelo Google Colab quanto local no Jupyter Notebook, que deve ser instalado na máquina do aluno, o que pode propiciar uma maior variedade de recursos. Os alunos também podem realizar eventuais mudanças no notebook, testando outras configurações, funcionalidades ou coletando novos dados.

Por fim (3) o resultado do experimento não se limita apenas ao momento da aula. Após o contato do aluno com esses novos conceitos e tecnologias, ele também pode utilizá-los em projetos próprios, como por exemplo em uma Iniciação Científica ou em um Trabalho de Conclusão de Curso, integrando dispositivos IoT como o Raspberry Pi.

V. RESULTADOS

O notebook foi organizado em 8 passos, sendo 1 etapa de preparação, 1 etapa de leitura e 6 atividades práticas:

- 1) Introdução aos Parâmetros *Chroma*;
- 2) Processamento do Áudio de uma Música;
- 3) Visualização dos Parâmetros Obtidos;
- 4) Instalação e Criação de Conta no Aplicativo Blynk;

- 5) Criação de um Projeto no Blynk;
- 6) Configuração do Aplicativo Blynk;
- 7) Interação com a Plataforma IoT por meio de Requisição HTTP.

Os seguintes Objetivos de Aprendizado (OA) foram estabelecidos para o experimento:

- **OA1:** Após ler as informações e assistir aos vídeos do Passo 1, os alunos serão capazes de descrever o que são parâmetros *Chroma*, e como eles podem ser usados para diferenciar músicas.
- **OA2:** Após a realização das atividades dos Passos 2 e 3, os alunos serão capazes de usar a biblioteca *pyAudioAnalysis* para processar áudios, demonstrando o processamento de áudios diferentes para professores e monitores, por meio de interação com o notebook Python.
- **OA3:** Após a realização das atividades dos Passos 4, 5, 6 e 7 os alunos serão capazes de enviar os parâmetros obtidos com o processamento do áudio para seu celular e obter esta mesma informação usando requisições HTTP em uma plataforma IoT.

Como requisitos, além dos arquivos presentes no Github da ferramenta, são necessários um computador com acesso à Internet e um celular iOS ou Android. Os 8 arquivos de músicas de gêneros diferentes para uso livre foram obtidos no portal Free Music Archive¹, conforme detalhado na Tabela I. Estes arquivos são obtidos pelos alunos a partir de um repositório público onde as músicas foram disponibilizadas por meio do uso da biblioteca *gdown*, conforme pode ser observado na Figura 3.

```
[ ] import gdown
output = "pop.wav"
id = "1StGRqIAqA18KkvTsT0Rb_8C8PqIF7YmM"
url = "https://drive.google.com/uc?id=" + id
gdown.download(url, output, quiet=False)

Downloading...
From: https://drive.google.com/uc?id=1StGRqIAqA18KkvTsT0Rb_8C8PqIF7YmM
To: /content/pop.wav
28.9MB [00:00, 44.2MB/s]
pop.wav
```

Observe que o arquivo deverá constar na aba lateral esquerda, conforme figura a seguir, onde foi baixado o arquivo "pop.wav"

Fig. 3. Obtenção de arquivos em Notebook no Google Colab

Após a obtenção dos arquivos e a instalação das bibliotecas necessárias de forma automatizada no ambiente *cloud* Google Colab, o primeiro passo consiste em assistir a um vídeo sobre a teoria relacionada, e observar os parâmetros que podem ser obtidos com o uso da biblioteca *pyAudioAnalysis*.

Os parâmetros usados na função *feature_extraction()* são a taxa de amostragem de 44100 Hz, tamanho da janela deslizante de 1024 amostras e o passo da janela de 128 amostras, além da opção ativa de que os parâmetros do tipo *delta* não devem ser obtidos. A ideia é que estes parâmetros possam ser ajustados de acordo com a necessidade do processamento dos áudios pelos alunos ou professores. Por exemplo, foi utilizada

¹<https://freemusicarchive.org/>

TABELA I
MÚSICAS PARA USO LIVRE UTILIZADAS NA VALIDAÇÃO DO NOTEBOOK PYTHON

Arquivo	Música	Banda	Tamanho (MB)	Duração (m:s)	Processamento (s)
blues.wav	Ashes Of Paradise	Pierce Murphy	46	04:35	180,9 ± 1,83
classical.wav	Jerry's Back	Crowander	30	02:56	114,88 ± 0,25
country.wav	Verona - Intro (ID 1407)	Lobo Loco	16	01:32	61,2 ± 0,45
eletronic.wav	Alternate	Xylo Ziko	39	03:54	154,71 ± 0,66
hiphop.wav	Dogstarmegalazer	Eaters	43	04:13	166,15 ± 0,38
jazz.wav	Among The Stars	Pierce Murphy	45	04:29	177,26 ± 1,09
pop.wav	We Are One	Scott Holmes Music	28	02:44	109,15 ± 1,01
rock.wav	Jazz Hole	Blue Wave Theory	27	02:43	117,82 ± 0,62

a frequência de 44100 Hz pois é a amostragem de áudio de um CD, e o processamento padrão de somente um canal (Mono).

O tempo médio, obtido a partir de 10 experimentos para cada áudio, necessário para o processamento dos áudios utilizados para validação está descrito na Tabela I. Convém destacar que o escopo desta pesquisa é exploratória, desta forma este tempo de processamento é um resultado preliminar. A execução em múltiplas vezes para o cálculo de parâmetros estatísticos como média e desvio-padrão está prevista em trabalhos futuros. Um exemplo da visualização do vetor *numpy* obtido no passo 3 é apresentado na Figura 4.

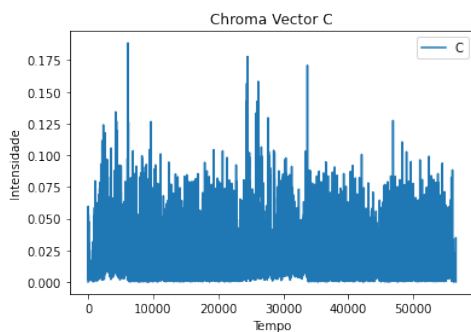


Fig. 4. Visualização dos parâmetros obtidos para o *Chroma C*

Os passos 4 e 5 detalham todos os procedimentos necessários para a criação de conta usando o aplicativo Blynk, e obtenção do *token de autenticação* para interação por meio de requisições HTTP no passo 7.

Como resultado, após a configuração do aplicativo conforme o passo 6, o aluno tem a interface da Figura 5, e pode realizar um teste inicial com uma requisição manual pelo seu navegador de Internet.



Fig. 5. Teste inicial com a interface criada pelo aluno no aplicativo Blynk

Após o passo 7, o aluno envia os parâmetros obtidos com o processamento do sinal de áudio para seu aplicativo por

meio da requisição HTTP, como pode ser observado na Figura 6. Estes mesmos parâmetros podem ser obtidos por meio de requisição HTTP para a plataforma IoT Blynk.



Fig. 6. Envio de parâmetros obtidos com o áudio para o Aplicativo Blynk

Há duas funcionalidades que são possíveis somente na execução local da ferramenta com o Jupyter Notebook, devido a problemas de compatibilidade entre bibliotecas no ambiente *cloud* Google Colab: execução da música conforme a Figura 7, e a visualização do cromagrama conforme a Figura 8, que é um exemplo do cromagrama para a música do gênero *blues*.

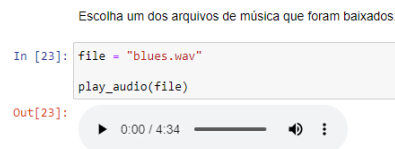


Fig. 7. *Player* e Chromagrama no Jupyter Notebook

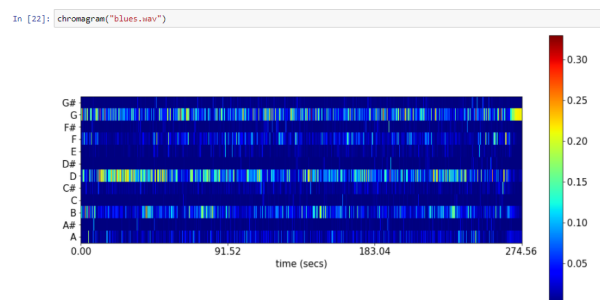


Fig. 8. Chromagrama no Jupyter Notebook em execução local da ferramenta

VI. TRABALHOS RELACIONADOS

O Curso Online Aberto e Massivo (MOOC, do inglês *Massive Open Online Course*) sobre processamento de sinais de áudio, oferecido pela Stanford na plataforma Coursera, apresenta atividades práticas com o programa Audacity, *Application Programming Interfaces* (APIs), bibliotecas em C++ e Python [6]. O MOOC disponível na plataforma Udemty também possui atividades práticas baseadas em Matlab [19]. Contudo, estes materiais estão em inglês.

JDSP é um ambiente de simulação para fomentar o aprendizado de estudantes sobre aspectos de processamento e codificação de sinais de fala, e pode ser usado por professores para demonstrar conceitos associados a aplicações que motivem os alunos, como os utilizados em celulares [1]. Contudo, o JDSP trata-se de uma solução independente sem integração com outros sistemas, enquanto a solução apresentada possui integração com uma interface *mobile* e uma plataforma IoT, o que pode reduzir a distância cognitiva entre a atividade prática realizada e sua possível aplicação prática.

SART3D é uma solução baseada em Matlab para facilitar a educação em processamento de sinais de áudio. Seu objetivo é possibilitar que estudantes experimentem algoritmos e obtenham seus resultados em tempo real. Foi validado com 45 estudantes por meio de um questionário sobre diferentes aspectos como habilidades, emoção, participação e desempenho. O uso da ferramenta aumentou o engajamento dos alunos, seu desempenho e entendimento sobre o assunto [5].

Outra solução baseada em Matlab é o AARAE (*Audio and Acoustical Response Analysis Environment*), que possui recursos visuais de interface do usuário, e que possibilitou que os alunos também ajudassem no desenvolvimento de alguns de seus módulos [4]. Um ponto negativo das ferramentas baseadas em Matlab é a dependência de um *software* proprietário, o que pode restringir sua acessibilidade.

Desta forma, observa-se um problema em aberto em relação à acessibilidade de materiais para atividades práticas para suportar a educação em processamento de sinais de áudio. Soluções encontradas na literatura estão em inglês (mesmo que as aulas possam ter legendas, os códigos e *softwares* associados podem não ter este suporte) e possuem dependência de *software* proprietário, o que restringe o público alvo.

O material desenvolvido está descrito em português, e foi organizado de acordo com Objetivos de Aprendizagem, para deixar claro aos estudantes o que irão aprender com determinadas atividades práticas, e ainda oferece integração com plataforma IoT para facilitar a integração da atividade prática com alguma aplicação do contexto do aluno, como um possível projeto. Esta ferramenta de código aberto pode ser executado não somente na nuvem da Google, mas também em um ambiente local.

O notebook apresentado pode ser testado no ambiente Google Colab², onde também é possível obter as músicas usadas para validação. É possível obter todo o código fonte no GitHub³, sob a licença *GNU General Public License v3.0*.

²https://colab.research.google.com/drive/1DgB5FgHltVHzJ41sLSwXmH_5q7hwmHR?usp=sharing

³<https://github.com/vthayashi/nb1-audioproc>

VII. CONCLUSÕES

Este trabalho apresentou uma ferramenta de código aberto para atividades práticas em processamento de sinais de áudio desenvolvida de acordo com os Objetivos de Aprendizagem em um Notebook Python que pode ser executado não somente na nuvem da Google, mas também em um ambiente local.

Como trabalhos futuros, pode-se integrar dispositivos para processamento do áudio em um dispositivo IoT. É possível realizar experimentos para estabelecer uma análise estatística do tempo de resposta, além da validação da proposta com alunos a partir da especificação de cenários, levantamento do perfil dos participantes e atividades prévias, e outros aspectos que podem ser obtidos por meio de questionários. A integração de outras ferramentas e bibliotecas como o "IPython.Display" e "LibROSA" pode expandir as funcionalidades da ferramenta.

REFERÊNCIAS

- [1] ATTI, V., AND SPANIAS, A. On-line simulation modules for teaching speech and audio compression techniques. In *33rd Annual Frontiers in Education, 2003. FIE 2003.* (2003), vol. 1, IEEE, pp. T4E–17.
- [2] BARTSCH, M. A., AND WAKEFIELD, G. H. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on multimedia* 7, 1 (2005), 96–104.
- [3] BLYNK. Blynk. <https://blynk.io/>, 2021. Acesso: 20/05/2021.
- [4] CABRERA, D., JIMENEZ, D., AND MARTENS, W. L. Audio and acoustical response analysis environment (aarae): a tool to support education and research in acoustics. In *INTER-NOISE and NOISE-CON congress and conference proceedings* (2014), vol. 249, Institute of Noise Control Engineering, pp. 1667–1676.
- [5] COBOS, M., AND ROGER, S. Sart3d: A matlab toolbox for spatial audio and signal processing education. *Computer Applications in Engineering Education* 27, 4 (2019), 971–985.
- [6] COURSERA. Audio signal processing course. <https://www.coursera.org/learn/audio-signal-processing>. Acesso: 28/05/2021.
- [7] FAUST, J. L., AND PAULSON, D. R. Active learning in the college classroom. *Journal on excellence in college teaching* 9, 2 (1998), 3–24.
- [8] GIANNAKOPOULOS, T. pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS one* 10, 12 (2015), e0144610.
- [9] GOOGLE. Google colab. <https://colab.research.google.com/>, 2021. Acesso: 20/05/2021.
- [10] HUITT, W. Bloom et al.'s taxonomy of the cognitive domain. *Educational psychology interactive* 22 (2011).
- [11] HYLÉN, J. Open educational resources: Opportunities and challenges. *Proceedings of open education* 4963 (2006).
- [12] JUPYTER, P. Jupyter notebook. <https://jupyter.org/>, 2021. Acesso: 20/05/2021.
- [13] KOKOTSAKI, D., MENZIES, V., AND WIGGINS, A. Project-based learning: A review of the literature. *Improving schools* 19, 3 (2016), 267–277.
- [14] KUROSE, J. F., AND ROSS, K. W. Redes de computadores e a internet. *São Paulo: Person* (2006), 28.
- [15] MADAKAM, S., LAKE, V., LAKE, V., LAKE, V., ET AL. Internet of things (iot): A literature review. *Journal of Computer and Communications* 3, 05 (2015), 164.
- [16] MÜLLER, M., AND EWERT, S. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR), 2011. hal-00727791, version 2-22 Oct 2012* (2011), Citeseer.
- [17] MÜLLER, M., KURTH, F., AND CLAUSEN, M. Audio matching via chroma-based statistical features. In *ISMIR* (2005), vol. 2005, p. 6.
- [18] SMITH, A. A. Teaching computer science to biologists and chemists, using jupyter notebooks: tutorial presentation. *Journal of Computing Sciences in Colleges* 32, 1 (2016), 126–128.
- [19] UDEMY. Learn audio processing like engineers do ! <https://www.udemy.com/course/learn-audio-processing-complete-engineers-course/>. Acesso: 28/05/2021.
- [20] WILSON, G., PEREZ, F., AND NORVIG, P. Teaching computing with the ipython notebook. In *Proceedings of the 45th ACM technical symposium on Computer science education* (2014), pp. 740–740.