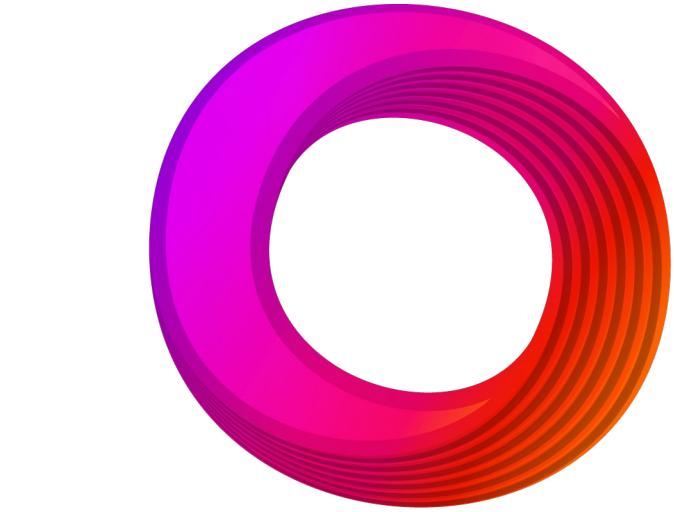
LUCIANO SANTOS DO SACRAMENTO



Cordaria

Desenvolvimento de Aplicação Web para Iniciação à Pratica do Violão e Guitarra

CORDARIA

Desenvolvimento de Aplicação Web para Iniciação à Pratica do Violão e Guitarra

Relatório final, apresentado ao público, como parte das exigências para prestação de contas deste objeto.

2021

Belo Horizonte / MG cordaria.com.br japraticouhoje@cordaria.com.br

RESUMO

O presente artigo visa pesquisar uma abordagem metodológica na prática do estudo de violão e guitarra, utilizando recursos disponíveis em plataformas webs. Pretende implementar uma aplicação cuja função é gerar exercícios práticos para as fases iniciais auxiliando assim, o professor e o educando.

Palavras-Chave: Educação musical; Programação web; Linguagens de programação.

ABSTRACT

This article aims to research a methodological approach in the practice of studying acoustic guitar and guitar, using resources available on web platforms. It intends to implement an application whose function is to generate practical exercises for the initial phases, thus helping the teacher and the student.

Key words: Music education; web programming; Programming languages.

Cordaria

SUMÁRIO

1 INTRODUÇÃO

A pesquisa “Cordaria – Desenvolvimento de aplicação web para iniciação à prática do violão e guitarra” surge em um momento delicado da história mundial: a pandemia do Coronavírus – COVID 19. Compelidos a este problema, os profissionais da área da educação, sobre tudo com atuação na área da música enfrentam grandes desafios para conseguir adequar suas atividades para o ensino a distância. Parte deste desafio é devido a escassez de recursos tecnológicos que fogem da metodologia tradicional de ensino, muita das vezes, limitada a vídeos, textos, imagens e de forma repetida. Os problemas se amplificam com usuários oriundos da classe baixa, que tem uma limitação considerável de hardware e/ou baixa conexão de internet.

2 PROBLEMA

Em busca de soluções para o lesionamento virtual, através de alguns contatos com educadores da área, foi percebido uma grande dificuldade na adaptação de muitos professores e alunos com a introdução repentina da prática do ensino a distância (EAD). Apesar das ferramentas disponíveis atualmente facilitarem o EAD em diversas áreas, na educação musical, sobretudo no ensino da sua prática, o cenário é diferente. Um dos problemas que destaco é devido ao modo de transmissão do som.

Tentarei abaixo, explicar de forma bem reduzida para facilitar a compreensão, como é feita a transmissão do áudio neste cenário:

1. O som emitido (que está em forma de onda analógica) é captada pelo microfone;
2. O microfone capta o nível de pressão sonora e a converte em impulso elétrico;
3. O sinal elétrico (analógico) é convertido em sinal digital, sendo processado em forma de arquivo(s);
4. Os arquivos são “quebrados” em pacotes e é transmitido do equipamento emissor para o equipamento receptor
5. O equipamento receptor assim que recebe, junta os pacotes e converte o sinal digital para analógico
6. O sinal analógico é enviado a saída de áudio e reproduzido em caixas acústicas ou fones de ouvido que transformam o sinal elétrico analógico em ondas sonoras.

Diferente de um texto ou uma imagem estática, os áudios (como também os vídeos), são arquivos volumosos e complexos, que, por sua vez, demoram mais para serem processados e transferidos, gerando um efeito chamado *Delay* (atraso). Tal efeito é bastante variável e leva em consideração a qualidade dos equipamentos do

transmissor e receptor, bem como do sinal de internet. Neste ambiente, o atraso do sinal prejudica na sincronização da prática de exercícios em conjunto aluno/professor, uma vez que a execução fica assíncrona, levando a muitos erros e dificuldades por parte do educando e prejudicando a avaliação do professor.

3 DESENVOLVIMENTO DA PESQUISA E DO MÉTODO

Utilizando meus conhecimentos de professor de violão (11 anos de experiência), quanto desenvolvedor web (3 anos), pretendo, através desta ótica, implementar uma aplicação web denominada “Cordaria” com a função de gerar exercícios básicos de digitação com foco na técnica. A finalidade dos exercícios é auxiliar o educando a executar as notas dadas através de uma escrita de tablatura, cuja a linguagem é de fácil entendimento comparado a escrita de partitura. O aplicativo ficará responsável por lhe informar o momento instantâneo da nota a ser tocada acompanhado da execução do áudio em um andamento pré-determinado. Desta forma, o aluno poderá praticar junto ao auxílio de um “correpetidor” virtual, em qualquer momento e lugar!

Para maior compatibilidade, a aplicação web será desenvolvida basicamente nas linguagens HTML, CSS e Java Script, além de bibliotecas para renderização de conteúdos visuais e execução de áudios, para que possa ser interpretada pela maioria dos navegadores reconhecidos, a exemplo o Chrome, Firefox e Safari, tanto em dispositivos de mesa quanto móveis, dispensando a instalação de outros aplicativos e a exigência de equipamentos mais robustos. Para o ambiente de desenvolvimento, utilizarei o software Virtual Studio Code como editor de códigos e os serviços de repositórios GitLab e GitHub para lidar com controle de versões, que estará disponível para acompanhamento mediante solicitação a futuros colaboradores.

O objeto final da pesquisa será disponibilizado em um servidor *web* por uma empresa contratada, bem como um domínio próprio do método, após a data de lançamento. O aplicativo inicialmente ficará com seu acesso disponível ao público de forma gratuita e irrestrito. Após o período de vencimento dos serviços adquiridos, tentaremos buscar outras fontes de recurso para a sustentabilidade do projeto.

Para manter as medidas vigentes de prevenção ao contágio e de enfrentamento da pandemia, todas as etapas desta pesquisa serão feitas *in loco*, de forma virtual e remota.

3.1 REFERENCIAIS

Em 2005, iniciei meus estudos teóricos em música e violão Clássico no conservatório da Fundação de Educação Artística (FEA). Lá, tive aulas com o violonista Teodomiro Goulart, do qual elaborou seu próprio método para lecionar. O Violar é um método de ensino de violão baseado em cartilhas para formular o exercício prático. Na minha opinião, um dos melhores que já vi, tanto é que utilizo até hoje com meus alunos. Com ele, podemos gerar muitas variações de exercícios apenas mudando a ordem das cartilhas. Desta forma, dispensa o uso de apostilas volumosas reduzindo o consumo de papel. O Violar me influenciou muito desde o momento que comecei a lecionar, tirando-me do senso comum e buscando outras formas de transmitir o conhecimento.
Desde então, comecei a desenvolver meus próprios métodos. Em alguns momentos de dificuldade em lecionar, desenvolvi um jogo básico de rítmica para o aprendizado da leitura e execução das figuras musicais.
Outro método que me incentivou a criar o Cordaria foi o software Rosetta Stone: Desenvolvido deste 1992,

> o Rosetta Stone é um software de ensino de línguas produzido pela Rosetta Stone Inc. O programa utiliza imagens, texto, som e vídeo para ensinar palavras e gramática por meio de repetição espaçada, sem tradução. A empresa chama este método de "Imersão Dinâmica" (um termo que eles têm registrado). (Roseta Stone – Wikipédia...,2021) https://pt.wikipedia.org/wiki/Rosetta_Stone.

Um dos recursos mais impressionantes do Rosetta é a capacidade de captar o áudio do microfone do educando, fazendo que ele responda as questões propostas de forma

oral. O sistema analisa o áudio gravado do aluno e faz uma comparação com o áudio de seu banco de dados, retornando assim, a avaliação o quanto e onde o falante acertou e onde errou.

3.2 ESBOÇO

Em 2013, A primeira versão de desenvolvimento do Cordaria (ainda sem nome na época) foi feita através do software PureData. Basicamente, o sistema consistia em gerar sequências de números pré-estabelecidos, baseado da tabela Midi e exibir em tela o resultado em um pentagrama. Este projeto foi logo descontinuado e não lançado, devido a exigência da máquina do usuário ter instalado e configurado adequadamente o software PureData e a não possibilidade do funcionamento on-line.

3.3 VERSÕES

3.3.1 Versão alfa (protótipo 1) até versão 0.6

Em 2017, comecei meus estudos de desenvolvimento web, do qual tive a oportunidade de desenvolver vários sites em Wordpress e, em 2020, devido a pandemia e a busca por novos métodos de ensino à música, para me adequar ao ambiente virtual, retomei o desenvolvimento deste projeto. O código foi reescrito na linguagem EcmaScript (conhecido popularmente como Java Script). Na ausência de um renderizador de partitura nativo desta linguagem, a saída de dados foi readequada a linguagem Tablatura. Apesar de não visualizarmos o resultado no pentagrama, este processo foi muito eficaz pois a tablatura é uma linguagem de fácil compreensão, diferente da

linguagem convencional, que exige conhecimentos prévios para sua leitura. Até este momento, não foi foi trabalhado a implementação sonora do sistemao. Evitei de entrar em detalhes da construção da lógica de programação até o momento, pois, como passei por várias linguagens de experimento e, do qual a especificidade das linguagens de programação, me fez ter vários algoritmos distintos, fugindo do escopo desta pesquisa. Descreverei os detalhes das próximas versões.

3.3.2 Versão Beta (protótipo 2) até versão 1.0.0

Com os resultados positivos da versão alfa, resolvi desenvolver o projeto Cordaria, do qual submeti e fui contemplado no Edital Lab promovido pela Secretaria de Estado da Cultura e Turismo, através da Lei Federal Aldir Blanc. A partir deste momento, possibilitou uma ampliação da equipe, do qual o layout e a identidade visual do sistema ficou a cargo do designer Gabriel Barreto.
Uma vez alcançado o objetivo de entender a lógica da linguagem, o próximo passo foi pensar no ambiente de frontend, uma vez que apenas com EcmaScript, o resultado era impresso em backend, enviado para o console. Para criar o frontend, optei por utilizar as linguagens EcmaScript, HTML e CSS, além do Bootstrap, para a responsividade entre as versões desktop e mobile. Para executar as tarefas referente a troca de cartas, foi utilizado a biblioteca Jquery e, para o áudio, foi utilizado a biblioteca Pizzicato.

3.3.2.1 Como Funciona o app?

Cada dígito representa qual dedo será pressionado e em qual casa, para gerar a nota exigida pelo exercício. Caso o dígito for igual a 0, a nota será dada pela corda solta.

QUADRO 1 – POSIÇÃO DOS DEDOS

0	1	2	3	4
Corda Solta Nenhum Dedo	Casa 1 Dedo Indicador	Casa 2 Dedo Médio	Casa 3 Dedo Anelar	Casa 4 Dedo Mindinho

Utilizando a lógica de programação orientada a objetos (POO), elegemos cada algarismo em um objeto, do qual foi denominado fragmento. Para trabalhar todos os dedos e corda solta em uma sequência, colocamos os 5 fragmentos possíveis em um objeto carta. Dada esta informação, a possibilidade do baralho completo, sem repetição, é de 5! (cinco fatorial) cujo o resultado é 120 cartas. Para diminuir a carga de processamento em gerar e randomizar todos os fragmentos a cada início de exercício, os dados do baralho são armazenados em um arquivo no formato JSON, que quando requisitados, apenas as cartas são embaralhadas.

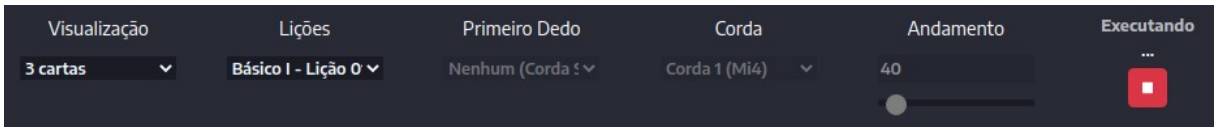
QUADRO 2 – ESTRUTURA BÁSICA DO ARQUIVO JSON - BARALHO

Baralho	Carta 0	Fragmento 0
		Fragmento 1
		Fragmento 2
		Fragmento 3
		Fragmento 4
	...	...
	Carta 119	...

Para iniciar o exercício, o usuário pode selecionar uma lição pré-definida, pensada para uma evolução progressiva, ou personalizá-la, caso o educando já tenha alguma experiência na prática. Cada lição contém o dedo inicial que começará cada carta, a corda que irá praticar, e o andamento (velocidade) das trocas de fragmentos. Uma vez selecionados estes parâmetros, o sistema requisita os dados do baralho, filtra as cartas

que começam com o dedo escolhido (4! = 24 cartas), as embaralham e armazenam em um novo objeto baralho, desta vez, em uma variável (memória) e não em um arquivo JSON.

FIGURA 1 – MENU DO SISTEMA



Os dados das lições pré-definidas também estão armazenados em arquivo JSON, do qual, quando a página de prática do sistema for aberta, é requisitado e armazenado em uma variável, interpretado e renderizado no navegador web do cliente.

QUADRO 3 - ESTRUTURA BÁSICA ARQUIVO JSON - LIÇÕES

Lições	Lição 1	Dedo Inicial	0
		Corda	6
		Andamento (BPM)	40
	...	...	..
	Lição 90	Dedo Inicial	4
		Corda	1
		Andamento (BPM)	200

3.3.2.2 Recursos de Áudio

A partir de agora, precisaremos nos preocupar com a parte auditiva do programa. Utilizaremos o software Sibelius (editoração e reprodutor de partitura) e sua biblioteca

de samples para emular o som do violão e assim, geraremos as notas e a exportaremos para o Logic Pro X (editor de áudio), do qual foram separadas contendo uma nota longa em cada arquivo. Para melhor performance, elencamos o formato MP3, com a configuração de *sample rate* em 44.100 hz e *bit rate* em 16 bits obtendo uma excelente compactação dos arquivos sem muita perda de qualidade. Através de um arquivo JSON, mapearemos cada arquivo de som e associaremos a suas respectivas notas. Pensando no reaproveitamento de código, esse arquivo será associado a um instrumento. Essa atitude possibilita, em futuras atualizações, a aplicação do método em outros instrumentos de corda, bastando apenas, a construção do banco de dados de áudio do novo instrumento e um seletor de escolha do mesmo.

QUADRO 4 – ESTRUTURA BÁSICA DO ARQUIVO JSON – GUITARRA

Corda: 1	Casa: 0	Valor: E5	Áudio: 10.mp3
	Casa: 1	Valor: F5	Áudio: 11.mp3
	Casa: 2	Valor: F#5	Áudio: 12.mp3
	Casa: 3	Valor: G5	Áudio: 13.mp3
	Casa: 4	Valor: G#5	Áudio: 14.mp3
..	...	...	...
Corda: 6	...	...	...

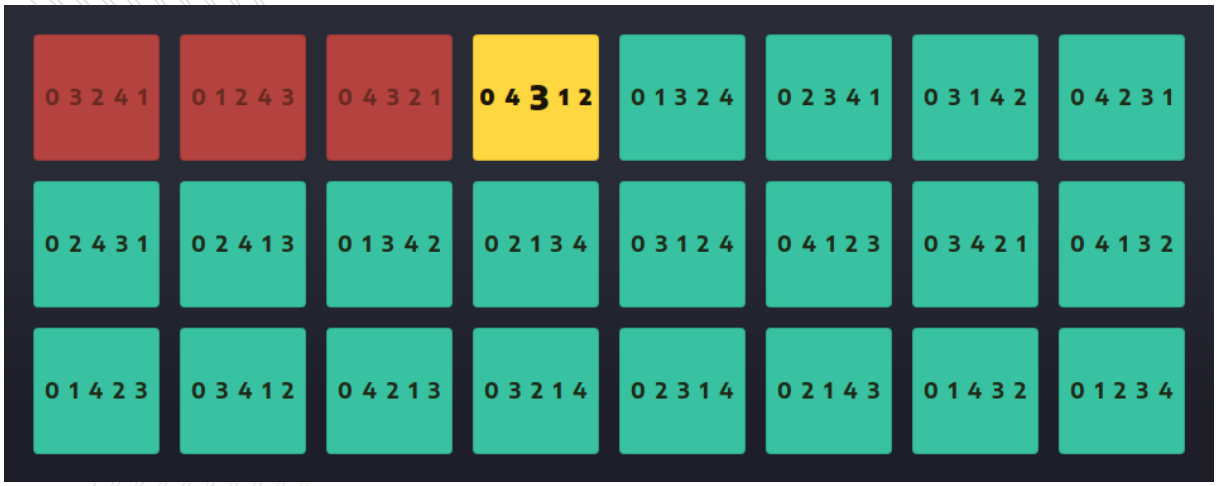
Devido a problemas verificados na na versão anterior de sobrecarga no processamento da renderização do conteúdo visual e auditivo simultaneamente comprometendo a sincronização destes elementos, foi necessário a substituição da biblioteca Pizzicato para Tone.js. O motivo pela troca é que, enquanto o Pizzicato renderiza nota a nota em tempo real de execução, o Tone.js nos possibilita a criação de sequências de samples para ser executado em um momento posterior, realizando tal tarefa de forma assíncrona. Dessa forma, o sistema cria uma *playlist* a partir da ordem das cartas já embaralhadas e a duração de cada evento sonoro. Uma vez criado a

sequência, prometemos a execução do áudio apenas quando o sistema estiver pronto para renderizar o início da prática, resolvendo o problema de dessincronização.

3.3.2.3 Modos de Visualização

O sistema possibilita dois modos de visualização das cartas:
No modo clássico, o sistema distribui todas as cartas na tela ordenada como na escrita ocidental, (da esquerda para direita e de cima para baixo, nesta ordem). Assim, cada fragmento da carta será destacado e a execução sonora da nota equivalente. O processo se repete até que o último fragmento da última carta for executado, momento este, final do exercício.

FIGURA 2 – MODO VISUALIZAÇÃO DE TODAS AS CARTAS



Já no modo três cartas, foi desenvolvido pensando na possibilidade de utilização do método em dispositivos móveis, estes, que por razão da sua tela reduzida, pode ficar difícil a visualização de componentes pequenos. Este modo consiste em uma visualização dinâmica, pois temos apenas as cartas atual (em amarelo) e vizinhas

(última carta que já foi executada em vermelho e futura carta em verde) sendo exibidas. Uma vez que acabamos de ler o último fragmento da carta atual, estes dados são escritos na tela de visualização da carta passada, a carta atual recebe os dados que estavam na tela de visualização da futura carta e a futura carta pega os dados (se houver) da próxima carta na sequência.

FIGURA 3 – MODO VISUALIZAÇÃO DE 3 CARTAS

3.3.3 Versão Oficial (estável) versão 1.1.0

Após o lançamento da versão 1.0.0, foi percebido um problema de sincronização entre o movimento das cartas e a execução do áudio. Neste momento percebi que, com apenas as linguagens “puras” junto as bibliotecas utilizadas exigiam um grande processamento de dados para que as tarefas executem em tempo real. Neste momento, tivemos que transcrever novamente o código em um framework que trabalhe de forma reativa, do qual foi elencado o Vue em sua versão 3 (em modo SPA).

3.3.4 Versão 1.1.2 (Atual)

Nesta versão, fizemos uma adaptação para a troca de framework puramente em Vue para um mais recente, o Nuxt. Tal escolha foi baseado no aproveitamento de código, já que o Nuxt tem incorporado a linguagem do Vue. Em termos de estrutura da aplicação na versão anterior, nada foi mudado. Com esta troca, possibilitou a maior visibilidade e a possibilidade do método de ser indexado pelos mecanismos de busca, uma vez que o aplicativo trabalha em modo universal ao invés de SPA.

3.4 PROJETOS FUTUROS – ATUALIZAÇÕES E MELHORIAS

Dentre os processos de atualizações e melhorias, podemos elencar:

- Ampliação do método para novos instrumentos de corda como cavaquinho, baixo e Ukulelê.
- Construção de novos algoritmos e exercícios mais elaborados que trabalham com múltiplas cordas e ampliação da tessitura dos instrumentos, para treino em outras regiões do braço para além da primeira posição;
- Criação de um aplicativo mobile em uma versão “baixável” que trabalhe em modo *off-line*;
- Desenvolvimento de um sistema de cadastro de usuários para acompanhamento e registro de lições executadas;
- Desenvolvimento de um sistema onde possa avaliar a performance do usuário.

4 CONCLUSÃO

O objetivo fundamental desta pesquisa foi concluído de forma satisfatória e o resultado está hospedado no sítio https://cordaria.com.br desde meados deste ano (2021). Neste link, encontraremos também outras informações como a introdução sobre o que é o Cordaria, o resumo e link para download desta pesquisa, o tutorial de uso com fotos e vídeos além de dicas de estudo para um melhor rendimento. Por estar no ar por um período muito curto, ainda não temos uma métrica da usabilidade do sistema por terceiros. Salientamos que, apesar dessa tarefa não está no escopo desta pesquisa, será executada em momento posterior, dando continuidade no desenvolvimento da aplicação.

Acreditamos que a aplicação web para iniciação à pratica do violão e guitarra possa impactar o cenário atual propondo e incentivando o desenvolvimento de novas abordagens desses conteúdos. Para além de professores e educandos, esperamos que esta pesquisa também possa chegar a programadores interessados na área, para que, de fato, possa vir a colaborar com atualizações de metodologias focadas ao fazer musical.

Deixamos a reflexão que o ensino da arte, bem como as demais áreas, não pode ser substituída em sua totalidade por uma máquina, já que a educação é a base para construção de uma sociedade. Contudo, o *e-learning* pode-se tornar muito eficaz na construção de talentos artísticos mesmo em momentos e que a ausência do instrutor for necessária.

Cordaria

5 ANEXO A – FLUXO DE FUNÇÕES DE CÓDIGO

```
// 1. Carregando as informações do exercício proposto pelo usuário.
load()
{
    getAudios() // Requisita todos os áudios.
    startTraining()  // Inicia o módulo de treinamento.
    {

// 2. Gerando dados do exercício
        generateExercise() // Embaralha, filtra  as cartas com o dedo inicial proposto e as armazenam.
        {
            sortIndex() // função auxiliar para ramdomizar baralho.
        }

// 3. Funções para lidar com áudio
        convertBpmToMs() // Converte as batidas por minuto em milissegundos.
        calculateRelease() // Calcula a proporção de "legato" das notas do banco de áudio.
        generateSequence() // Gera a sequência ordenada de notas com o andamento  propostos.
        {
            getNotes(fret) // Requisita a nota proposta pelo framento da carta para gerar a sequência.
        }
        Play() // Inicia a execução (animação) do jogo através de um temporizador no andamento proposto.
        {
            sequence.start() // Inicia a execução da sequência do áudio.

// 4. Funções para lidar com animação do conteúdo visual.
            startAnimateCards() // Inicia a execução visual e animação cartas.
            startAnimateValues() // Inicia a execução visual e animação dos valores das cartas.
        }
    }
}

// 5. Interrupção e Reinicio do jogo.
destroyed()
{
    stop() // Interrompe o jogo e atualiza a página da prática ao quando o usuário clicar no botão stop.
}
```

REFERÊNCIAS

AWESOME, Fort. Fort Awesome, 2021. Disponível em: <https://fortawesome.com/>. Acesso em 30 jun. de 2021.

BOOTSTRAP. Bootstrap, 2021. Disponível em: <https://getbootstrap.com/>. Acesso em 30 jun. de 2021. MOZILLA. MDN Web Docs, 2021. Disponível em: <https://developer.mozilla.org/pt-BR/>. Acesso em 30 jun. de 2021.

CHOI, Kyuwoo. Vue-Slick-Carousel, 2021. Disponível em <https://www.npmjs.com/package/vue-slick-carousel>. Acesso em 30 jun. de 2021.

COD3R. Curso Web Moderno com JavaScript 2021 + Projetos, 2021. Disponível em: <https://www.udemy.com/course/curso-web/learn/lecture/8774806#overview>. Acesso em 30 jun. de 2021.

FOUDATION, OpenJS. JQuery, 2021. Disponível em: <https://jquery.com/>. Acesso em 30 jun. de 2021.

GOOGLE. Google Fonts, 2021. Disponível em: <https://fonts.google.com/>. Acesso em 30 jun. de 2021.

GUILLEN, Alejandro M et al. Pizzicato, 2021. Disponível em: <https://github.com/alemangui/pizzicato>. Acesso em 30 jun. de 2021.

LABS, Nuxt. Nuxt – The Intuitive Vue Framework, 2021. Disponível em: <https://nuxtjs.com>. Acesso em 30 jun. de 2021.

MANN, Yotam. Tone.js, 2021. Disponível em: <https://tonejs.github.io/>. Acesso em 30 jun. de 2021.

MONTE, Limon. Sweetalert2, 2021. Disponível em: <https://github.com/sweetalert2/sweetalert2>. Acesso em 30 jun. de 2021.

MOZILLA. MDN Web Docs, 2021. Disponível em: <https://developer.mozilla.org/pt-BR/>. Acesso em 30 jun. de 2021.

PENNEC, Nicolas e PARSA, Pooya. Nuxt/Sitemap, 2021. Disponível em: <https://sitemap.nuxtjs.org/>. Acesso em 30 jun. de 2021.

Roseta Stone – Wikipédia..., 2021. Disponível em: <https://pt.wikipedia.org/wiki/Rosetta_Stone>. Acesso em 30 jun. de 2021.

TEAM, Core. BootstrapVue, 2021. Disponível em: <https://bootstrap-vue.org/>. Acesso em 30 jun. de 2021.

YOU, Evan. Vue.js, 2021. Disponível em: <https://vuejs.org>. Acesso em 30 jun. de 2021.

Cordaria

Esta pesquisa foi possível devido ao apoio da Lei Emergencial de Incentivo à Cultura – Lei Aldir Blanc, Governo Federal, Ministério do Turismo e Secretaria Especial de Cultura, através da aprovação no Edital LAB no 14/2020 – “Seleção de Bolsistas para as Áreas Artísticas Técnicas e de Produção Cultural” promovido pela Secretaria de Cultura e Turismo de Minas Gerais – SECULT.

CULTURA E TURISMO MINAS GERAIS GOVERNO DIFERENTE. ESTADO EFICIENTE.

SECRETARIA ESPECIAL DA CULTURA MINISTÉRIO DO TURISMO

PÁTRIA AMADA BRASIL GOVERNO FEDERAL