

Tércio Souza de Jesus

**Chordix: um Aplicativo para
Auto-Acompanhamento Musical Através de
Cifras**

Salvador

2017

Tércio Souza de Jesus

Chordix: um Aplicativo para Auto-Acompanhamento Musical Através de Cifras

Monografia apresentada ao Curso de graduação em Ciência da Computação, Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal da Bahia – UFBA

Instituto de Matemática e Estatística

Departamento de Ciência da Computação

Orientador: Rodrigo Rocha Gomes e Souza

Salvador

2017

Tércio Souza de Jesus

Chordix: um Aplicativo para Auto-Acompanhamento Musical Através de Cifras/
Tércio Souza de Jesus. – Salvador, 2017-
52p. : il. (algumas color.) ; 30 cm.

Orientador: Rodrigo Rocha Gomes e Souza

Monografia – Universidade Federal da Bahia – UFBA
Instituto de Matemática e Estatística
Departamento de Ciência da Computação, 2017.

1. computação musical. 2. instrumentação vocal. 2. cifras. I. Rodrigo Rocha
Gomes e Souza. II. Universidade Federal da Bahia. III. Instituto de Matemática e
Estatística. IV. Chordix: um Aplicativo para Auto-Acompanhamento Musical Através
de Cifras

Tércio Souza de Jesus

Chordix: um Aplicativo para Auto-Acompanhamento Musical Através de Cifras

Monografia apresentada ao Curso de graduação em Ciência da Computação, Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Salvador, 10 de abril de 2017:

Rodrigo Rocha Gomes e Souza
Orientador

Sandro Santos Andrade
Convidado 1

Jean Clemisson Santos Rosa
Convidado 2

Salvador
2017

Agradecimentos

Começo agradecendo a Deus pelo dom da Vida. Sem Ele, certamente não estaria aqui. O meu muito obrigado especialmente aos meus pais por terem renunciado tantas coisas em favor do meu crescimento, e por sempre acreditarem em mim, inclusive quando eu não o fazia. À minha irmã, Thais Souza de Jesus, o meu reconhecimento pela incansável paciência comigo. Seu apoio foi sem dúvida essencial para que eu conseguisse chegar até aqui. Agradeço à minha namorada, Larissa Mascarenhas de Oliveira, por ser o presente e ser sempre presente em minha vida. Obrigado por todo o seu companheirismo e amor. A todos os meus amigos que são sempre presentes, sejam nos bons ou maus momentos, eu gostaria de falar que essa amizade tornou a minha jornada muito mais fácil e agradável. Agradeço à Universidade Federal da Bahia – UFBA, por todas as oportunidades que me foram dadas, bem como ao meu orientador Rodrigo Rocha Gomes e Souza, que com toda a sua paciência me guiou e acreditou neste projeto junto comigo. Aos mestres e professores que compartilharam suas histórias e vasto conhecimento com muito carinho, fica um agradecimento que não consigo transpor em palavras. Parte do meu mérito também pertence a vocês.

*“Mas Deus prova o seu amor para conosco,
em que Cristo morreu por nós,
sendo nós ainda pecadores.
(Bíblia Sagrada, Romanos 5, 8)*

Resumo

A tecnologia digital tem crescido como um suporte no universo da música, tornando-se uma aliada para o desenvolvimento e aprendizagem de cantores e instrumentistas. Porém, casos em que cantores possuem dificuldades para aprender a tocar um instrumento visando o seu aprimoramento vocal, indisponibilidade de músicos instrumentistas para o auxílio ou a falta de acompanhamento instrumental para seus ensaios, podem acabar resultando no baixo progresso do aprendizado das técnicas vocais. Soluções como instrumentos virtuais que são desenvolvidos envolvendo tecnologia e música são capazes de suprir a falta destes recursos, beneficiando os cantores ao permitirem realizar um trabalho de auto-acompanhamento vocal através da computação musical. Portanto, este projeto relata as etapas do desenvolvimento de um aplicativo multiplataforma de auto-acompanhamento musical, descrevendo conceitos de teoria musical e ferramentas utilizadas permitindo os usuários executarem acordes musicais a partir de uma cifra musical em formato *HTML*. Por fim, testes foram realizados sobre a aplicação a fim de avaliar a compatibilidade com os *browsers* de navegação, tempo de resposta ao tocar os acordes e a qualidade sonora.

Palavras-chave: computação musical, plugins de áudio, cifras, multiplataforma, instrumentação vocal.

Abstract

Digital technology has grown as a support in the universe of music, becoming an ally for the development and learning of singers and instrumentalists. However, in cases where singers have difficulties to learn to play an instrument aiming at their vocal improvement, the unavailability of instrumental musicians for the aid or the lack of instrumental accompaniment for their rehearsals may end up resulting in the low progress of the vocal techniques. Solutions such as virtual instruments that are developed involving technology and music are able to overcome the lack of these resources, benefiting singers by allowing them to perform a self-accompaniment work through music computation. Therefore, this project reports on the steps of developing a multiplatform musical self-monitoring application, describing concepts of musical theory and tools used allowing users to perform musical chords from a musical chord in HTML format. Finally, tests were carried out on the application in order to evaluate the compatibility with navigation browsers, response time when playing chords and sound quality.

Keywords: musical computation, audio plugins, chords, multiplatform, vocal instrumentation.

Lista de ilustrações

Figura 1 – Intervalos musicais	26
Figura 2 – Intervalos musicais em um semitom da esquerda para a direita	27
Figura 3 – Intervalos musicais da direita para a esquerda em um semitom	27
Figura 4 – Monocórdio criado por Pitágoras	28
Figura 5 – Corda esticada em estado de vibração	28
Figura 6 – Corda esticada com 03 extremidades	28
Figura 7 – Oitavas em um teclado	29
Figura 8 – Comparação entre diagrama de acordes e braço de uma guitarra	31
Figura 9 – Diagrama do C	32
Figura 10 – Diagrama do Cm	33
Figura 11 – Parte da cifra “Redemption Song - Bob Marley”	33
Figura 12 – Fluxo de trabalho da <i>Web Audio API</i>	34
Figura 13 – Implementação para tocar um áudio com a ferramenta <i>Web Audio API</i>	35
Figura 14 – Simples implementação do <i>TSW</i>	35
Figura 15 – Utilização do <i>pedalboard.js</i>	37
Figura 16 – Utilização do <i>Tone.js</i>	37
Figura 17 – Interface do GarageBand	38
Figura 18 – Listagem de cifras no Decifra-me	39
Figura 19 – Tela inicial	41
Figura 20 – Cifra carregada	42
Figura 21 – Simples modelo cliente-servidor do Chordix	44
Figura 22 – Estrutura do acorde	45

Lista de tabelas

Tabela 1 – Escalas musicais maiores	29
Tabela 2 – Escala menor das 12 notas naturais	30

Lista de abreviaturas e siglas

DMI	Digital Musical Instruments
HTML	HyperText Markup Language
API	Application Programming Interface
TSW	Theresa's Sound World
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol

Lista de símbolos

b	Bemol
#	Sostenido

Sumário

1	INTRODUÇÃO	23
1.1	Objetivos do trabalho	24
2	FUNDAMENTAÇÃO TEÓRICA	25
2.1	Teoria musical	25
2.1.1	Notas musicais	25
2.1.2	Características das Notas Musicais	26
2.1.3	Intervalos musicais	26
2.1.4	Escalas musicais	27
2.1.5	Acordes musicais	29
2.1.6	Tríades maiores e menores	30
2.1.7	Formação de acordes no violão	31
2.1.8	Cifras	32
2.2	Web Audio API	33
2.2.1	Bibliotecas de áudio	35
2.2.2	Theresa's Sound World	35
2.2.3	Pedalboard.js	36
2.2.4	Tone.js	36
2.3	Trabalhos correlatos	37
2.3.1	GarageBand	37
2.3.2	Decifra-me	38
3	DESENVOLVIMENTO DO APLICATIVO	41
3.1	Objetivo	41
3.2	Funcionamento	41
3.2.1	Desenvolvimento	43
3.2.2	Tecnologias Utilizadas	43
3.2.3	Web Service	44
3.2.4	Extração e Identificação	44
3.2.5	Execução dos acordes	46
4	AVALIAÇÃO	47
5	CONCLUSÃO	49
5.1	Trabalhos Futuros	49

REFERÊNCIAS 51

1 Introdução

Com a crescente evolução da tecnologia, cada vez mais novas aplicações vêm sendo desenvolvidas para atender as necessidades dos usuários. As modernas aplicações para as plataformas *mobile* e *desktop* tendem a facilitar a maneira como as pessoas interagem com o mundo real. Porém, é cada vez mais frequente a migração das aplicações dos dispositivos estacionários para os dispositivos móveis, auxiliando os usuários à medida que o acesso se dá de forma rápida e independentemente do lugar (WEITGENANT, 2014).

Assim, com a crescente evolução da tecnologia da informação, aplicações voltadas para o universo da música foram sendo desenvolvidas com o surgimento de instrumentos musicais virtuais (WEITGENANT, 2014). Desta forma, novas idéias de instrumentos musicais surgiram com o objetivo de aproveitar os avanços da informática e prover interações musicais de uma maneira mais fácil aos usuários.

Logo, através das tecnologias digitais tem-se o surgimento dos Instrumentos Musicais Digitais, conhecido pela sigla *DMI*, do inglês *Digital Musical Instruments*, sendo instrumentos compostos de uma interface que envia sinais ou sons e são controlados por um sistema computadorizado (PATRICIO, 2012). Essa tecnologia serviu de base para o crescimento de aplicações, principalmente em dispositivos móveis, por conter um fácil acesso e uma melhor interação musical, sendo sensível ao toque, o que não tem a mesma sensibilidade em computadores pessoais, utilizando-se dos teclados por exemplo.

Barreiras podem dificultar o aprendizado em algum tipo de instrumento musical ou o desenvolvimento de técnicas de canto. Logo, para garantir um bom desenvolvimento na parte técnica de um instrumento quanto à instrumentação vocal, obter conhecimentos musicais em noções de ritmo, melodia e harmonia são elementos essenciais para o progresso desse *hobby*, que além disso, pode vir a se tornar algo profissional.

A música é a forma de expressão mais antiga da raça humana, tendo a voz como o primeiro instrumento musical (ALVARENGA, 2013). Assim, pessoas que preferem seguir com esta opção não precisam necessariamente ter conhecimentos técnicos nos mais variados tipos de instrumentos musicais para o autodesenvolvimento. Neste caso, diferentes recursos como: alguém disponível que saiba tocar algum instrumento ou playback de fundo, música de fundo previamente gravada, são suficientes para iniciar o processo de desenvolvimento do instrumento vocal.

Portanto, um item bastante requisitado nesse processo, sendo utilizado pela maioria dos músicos e cantores, que podem ser encontrados facilmente nos mais variados sites do gênero musical, são as cifras musicais que podem ser representadas de muitas formas. A forma mais comum de representar esse sistema de notação musical é a junção de estrofes

da música e letras que indicam os acordes a serem executados.

Assim sendo, desenvolveu-se um aplicativo para as plataformas *desktop* e dispositivos móveis que permite ao usuário executar acordes musicais a partir de uma cifra *HTML* para o auto acompanhamento musical.

1.1 Objetivos do trabalho

- A Realizar pesquisas de algumas bibliotecas de áudio que permitam emitir sons característicos de instrumentos musicais a partir dos acordes. Para a utilização dessas bibliotecas devem ser levadas em consideração a qualidade da documentação para uma boa implementação dos códigos, a proximidade dos sons emitidos com as dos instrumentos reais e o bom funcionamento em diversas plataformas;
- B Fazer o reconhecimento e extração dos acordes em uma cifra após o estudo de como as mesmas são estruturadas e comportadas em meio as estrofes da música ao longo da página *HTML*. Após a análise, usaremos uma linguagem de programação interpretada, *JavaScript* que uma de suas principais responsabilidades neste projeto, será a varredura em torno do arquivo para a identificação dos acordes e estrofes;
- C Permitir aos usuários uma boa visualização das estrofes e botões para as execuções dos acordes, levando em consideração uma característica bastante crucial que determinará o bom desenvolvimento da aplicação, sendo a latência que será a medição do tempo de retorno entre a ação ao se executar um acorde por meio de um botão e o tempo de resposta do som deverá ser extremamente baixa, garantindo uma boa usabilidade. Assim, técnicas de implementação para minimizar esse efeito deverão ser pesquisadas para garantir as execuções e o retorno ser de forma instantânea, provendo uma maior percepção;
- D Adaptar o *layout* da aplicação para as multiplataformas como *desktop* e dispositivos móveis.

2 Fundamentação teórica

A seção 2.1 descreve os conceitos relacionados à teoria musical. A seção 2.2 explica o funcionamento de áudios na *Internet* bem como o seu desenvolvimento no cenário atual. Por fim, na seção 2.3 são apresentados os trabalhos correlatos.

2.1 Teoria musical

Segundo (PRIORE, 2015), a teoria musical é ensinada na maioria das universidades de música abordando os conceitos de intervalos musicais, reconhecimento de harmonia, aperfeiçoamento dos ditados rítmicos, harmônico e melódico. Essa área de conhecimento descreve elementos da música e inclui o desenvolvimento e aplicação de métodos para a composição e análise de músicas através de notações (SFETCU, 2014).

2.1.1 Notas musicais

A música assumiu uma grande importância na época da Idade Média entre os clérigos. Essa característica de ascensão é explicada pelo fato de os monges daquela época possuírem tempo e oportunidade em relação ao conhecimento musical através das bibliotecas dos mosteiros provenientes da civilização clássica, bem como das liturgias que povoavam as cerimônias religiosas.

Considerado o pai das notas musicais e organizador desse sistema de notação musical conhecido até os dias atuais, Guido de Arezzo (992 - 1050)¹ batizou-as proveniente dos estudos da construção de uma escala musical simplificada para facilitar o aprendizado e compreensão dos alunos de uma peça musical.

Através da solmização, sistema que atribui uma sílaba distinta a cada nota em uma escala musical, Arezzo criou a maioria das notas musicais ao extrair do texto de um hino a São João Batista composto por Paulo Diácono (séc. VIII)², 06 sílabas das 06 frases iniciais do refrão, que eram cantadas os seguintes versos em latim:

Ut queant laxis, **R**esonare fibris, **M**ira gestorum, **F**amuli tuorum, **S**olve
polluti, **L**abii reatum.

A sílaba **si** foi incluída posteriormente como homenagem a Sante Iohannes, São João, através das iniciais do nome e sobrenome, bem como a alteração de **ut** para **do** por

¹ Biografias y Vidas. Guido d'Arezzo. Acessado em Março de 2017 de <<http://www.biografiasyvidas.com/biografia/g/guido.htm>>

² Treccani, la cultura italiana. Pàolo Diacono. Acessado em Março de 2017 de <<http://www.treccani.it/enciclopedia/paolo-diacono/>>

Giovanni Battista Doni (1594 - 1647)³. Posteriormente, as notas foram representadas como **C, D, E, F, G, A e B** sendo elas **dó, ré, mi, fá, sol, lá e si** respectivamente.

2.1.2 Características das Notas Musicais

A cada nota musical é associada uma frequência, o número de eventos que ocorrem em um determinado intervalo de tempo, que é a mesma frequência que ocorre na agitação das moléculas de ar a partir da vibração de uma corda de um violão por exemplo. Para cada frequência há um som, ou nota, diferente. Logo, segundo (KROGER, 2012), uma nota é um símbolo que representa um som musical que possui a combinação de frequência na unidade de *Hertz*, unidade de medida do SI - Sistema Internacional (SANTO, 2006) com o símbolo **Hz**, e o seu nome.

2.1.3 Intervalos musicais

As notas musicais que foram introduzidas na seção anterior são denominadas de notas musicais naturais, também conhecida como **notas diatônicas**. Assim sendo, entre as notas **C** e **D** por exemplo, existe um intervalo em que sua unidade de medida é dada em tons.

(KROGER, 2012) afirma que o acidente musical é representado por sinais que podem aumentar ou diminuir um tom ou mais semitons, sendo o semitom a unidade de medida de menor escala usada no sistema tonal ocidental. A representação do aumento e diminuição é dada a seguir:

- \flat : Símbolo que representa o decréscimo em um semitom, sendo chamado de bemol;
- \sharp : Símbolo que representa o aumento em um semitom, sendo chamado de sustenido.

As transições com os intervalos de tons (T) e semitons (ST) são mostrados na (Figura 1).

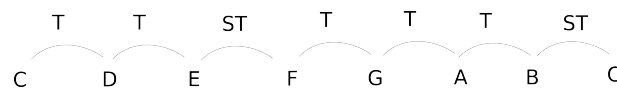


Figura 1 – Intervalos musicais

A partir da Figura 1, observa-se que mesmo a elevação sendo de um tom o mesmo não ocorre entre as notas de **E** para **F** e de **B** para **C**.

Para a elevação em um semitom, é visualizada na Figura 2

³ Biografías y Vidas. Giovanni Battista Doni. Acessado em Março de 2017 de <<http://www.biografiasyvidas.com/biografia/d/doni.htm>>

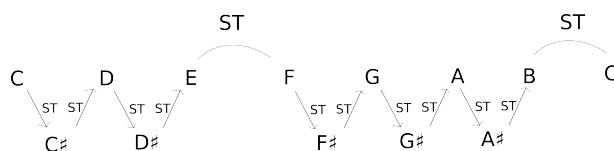


Figura 2 – Intervalos musicais em um semitom da esquerda para a direita

Através da Figura 2 representada, observa-se a presença de acidentes musicais conforme foi explicado. Logo, $C\sharp$ lê-se “dó sustenido” por exemplo.

Para o decréscimo da nota, é visualizado na Figura 3

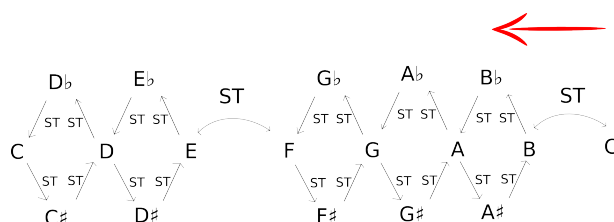


Figura 3 – Intervalos musicais da direita para a esquerda em um semitom

Obedecendo o sentido da leitura, observa-se o acidente musical caracterizado pelo símbolo b . A nota Db lê-se “ré bemol” por exemplo.

Observando a Figura 3, nota-se a presença de notas que ocupam o mesmo espaço, mas com nomes diferentes, caracterizando como notas **enarmônicas**, notas que possuem a mesma frequência, ou seja, sons, mas com grafias e nomes diferentes. São as seguintes notas enarmônicas elevadas ou decrescidas em semitom somando ao total de 12 notas:

- Db e $C\sharp$;
- Eb e $D\sharp$;
- Gb e $F\sharp$;
- Ab e $G\sharp$;
- Bb e $A\sharp$.

2.1.4 Escalas musicais

Um experimento realizado na Grécia Antiga protagonizado por Pitágoras (580-497 a.C.)⁴ foi a criação de um instrumento de uma única corda, denominada Monocórdio (ABDOUNUR, 2006). Seu experimento se deu a partir da observação de pesos diferentes dos martelos que eram batidos em uma ferraria, produziam sons diferentes (SOARES,

⁴ ebiografia. Pitágoras. Acessado em Março de 2017 de <<https://www.ebiografia.com/pitagoras/>>

2006). Assim, Pitágoras inventou um instrumento que se assemelha a um violão (Figura 4)⁵ com apenas uma corda.

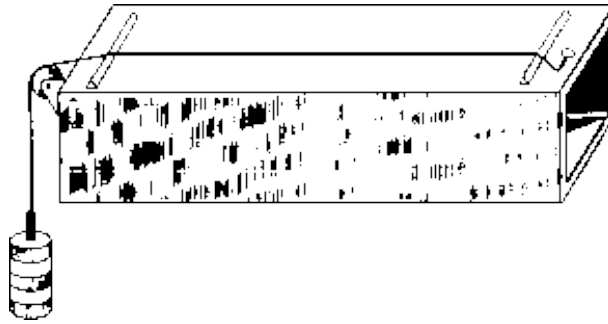


Figura 4 – Monocórdio criado por Pitágoras

Com a corda presa nas extremidades, ao tocá-la, a mesma entra em estado de vibração produzindo um som (Figura 5).



Figura 5 – Corda esticada em estado de vibração

Pitágoras então resolveu dividir em duas partes iguais a mesma corda (Figura 6)



Figura 6 – Corda esticada com 03 extremidades

Após a divisão da corda, Pitágoras percebeu que a tonalidade da corda era a mesma de quando possuam apenas 02 extremidades (Figura 5), porém em outra frequência. Por exemplo, imaginando-se que a corda estivesse emitindo um som característico à nota **A** na frequência 440 hz, após realizar a divisão, a mesma nota **A** poderia ser escutada em outra frequência, 880 hz, só que uma oitava acima, sendo a mesma tonalidade 8 notas à frente. Para uma melhor visualização, é mostrado um cenário de uma nota oitavada na Figura 7 (KROGER, 2012).

Assim sendo, na Figura 7 observa-se que o **C** mais à direita se repete após o **B**. A distância entre as duas notas iguais é dentro de um espaço de 08 notas, ou seja, o último **C** está uma oitava acima.

Pitágoras continuou com o seu experimento realizando mais divisões onde notava outros sons, ou seja, notas diferentes mas, que soavam de uma forma agradável. Dessa forma, foram surgindo as **escalas musicais** com as combinações das divisões.

⁵ Lucas Costa Soares, O Monocórdio de Pitágoras. Acessado em Março de 2017 de <<http://www.ghc.usp.br/server/Sites-HF/Lucas-Soares/monocordio%20de%20pitagoras.html>>

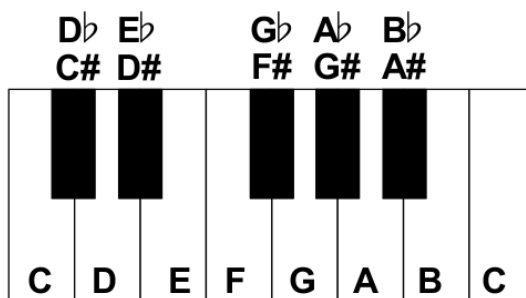


Figura 7 – Oitavas em um teclado

As escalas musicais são sequências de notas de maneira ordenada⁶ e possuem diferentes categorias como **escala maior natural**, **escala menor natural**, **diminuta**, **harmônica** e **melódica**. Porém, somente as escalas naturais **maior** e **menor** serão abordadas nesta seção.

A escala maior natural segue a seguinte sequência de tons: **tom, tom, semitom, tom, tom, tom, semitom** a partir de qualquer nota das 12 notas apresentadas. Na Tabela 1 é mostrada a sequência dos tons das notas fundamentais.

Tabela 1 – Escalas musicais maiores

Intervalos da escala	Tom	Tom	Semi-Tom	Tom	Tom	Tom	Semi-Tom	
Escala Dó maior	C	D	E	F	G	A	B	C
Escala Ré maior	D	E	F#	G	A	B	C#	D
Escala Mi maior	E	F#	G#	A	B	C#	D#	E
Escala Fá maior	F	G	A	Bb	C	D	E	F
Escala Sol maior	G	A	B	C	D	E	F#	G
Escala Lá maior	A	B	C#	D	E	F#	G#	A
Escala Si maior	B	C#	D#	E	F#	G#	A#	B

A mesma lógica é seguida com as demais 05 notas que já foram apresentadas.

A escala menor natural é formada a partir da sequência **tom, semitom, tom, tom, semitom, tom, tom** (Tabela 2). Também há a alteração do sustenido (♯) pelo bemol (b).

As notações que estão entre os parênteses são as notas finais.

2.1.5 Acordes musicais

Um acorde musical é a combinação de 03 ou mais notas diferentes soando de forma simultânea (BENWARD, 2003).

Existem muitas formas de se combinarem as notas para resultarem nos mais diversos acordes, onde cada um deles recebe um nome específico que são baseados nas

⁶ Descomplicando a música, Escalas Musicais. Acessado em Março de 2017 de <<http://www.descomplicandoamusic.com/escalas-musicais/>>

Tabela 2 – Escala menor das 12 notas naturais

ESCALA	NOTAS							
C menor	C	D	E \flat	F	G	A \flat	B \flat	C
D\flat menor	D \flat	E \flat	F \flat (E)	G \flat	A \flat	B $\flat\flat$ (A)	C \flat	D \flat
D menor	D	E	F	G	A	B \flat	C	D
E\flat menor	E \flat	F	G \flat	A \flat	B \flat	C \flat (B)	D \flat	E \flat
E menor	E	F \sharp	G	A	B	C	D	E
F menor	F	G	A \flat	B \flat	C	D \flat	E \flat	F
G\flat menor	G \flat	A \flat	B $\flat\flat$ (A)	C \flat (B)	D \flat	E $\flat\flat$ (D)	F \flat (E)	G \flat
G menor	G	A	B \flat	C	D	E \flat	F	G
A\flat menor	A \flat	B \flat	C \flat (B)	D \flat	E \flat	F \flat (E)	G \flat	A \flat
A menor	A	B	C	D	E	F	G	A
B\flat menor	B \flat	C	D \flat	E \flat	F	G \flat	A \flat	B \flat
B menor	B	C \sharp	D	E	F \sharp	G	A	B

notas fundamentais⁷.

Tomando por base as notas naturais “**dó, ré, mi, fá, sol, lá, si**”, alguns acordes recebem o mesmo nome das notações, sendo chamados de acordes naturais, em que cada um deles são formados por 03 notas, estabelecendo assim uma **tríade**.

Nesta seção serão abordados os conceitos e exemplos dos acordes **maiores** e **menores** bem como as suas formações em um instrumento do tipo violão. Porém, categorias de acordes como **suspenso**, **diminuto** e **alterados** fazem parte dos acordes musicais.

2.1.6 Tríades maiores e menores

Os acordes musicais maiores, ou **tríades maiores**, são formadas pelo primeiro, terceiro e quinto grau nas suas respectivas escalas. Ou seja, o acorde de **sol maior** a partir de sua escala “**G, A, B, C, D, E, F \sharp , G**” é composto pelas seguintes notas:

- **G**: sendo o 1º grau a própria nota do acorde, caracterizando como a tônica;
- **B**: 3º grau **maior**;
- **D**: 5º grau.

Para os acordes musicais menores, ou **tríades menores**, sua característica ocorre na mudança do 3º grau de maior para menor. Assim, para a composição de notas do acorde de **sol menor** a partir de sua escala “**G, A, B \flat , C, D, E \flat , F, G**” serão as seguintes notas

- **G**: sendo o 1º grau a própria nota do acorde, caracterizando como a tônica;
- **B \flat** : 3º grau **menor**;
- **D**: 5º grau.

⁷ Descomplicando a música, Acordes. Acessado em Março de 2017 de <<http://www.descomplicandoamusica.com/acorde/>>

Os acordes menores são acrescentados pela letra **m** em sua notação. Assim, a representação do acorde de **dó menor** é **Cm**.

2.1.7 Formação de acordes no violão

Na maioria das cifras musicais, os acordes são representados em diagramas que se assemelham à um violão ou guitarra por exemplo (Figura 8)⁸.

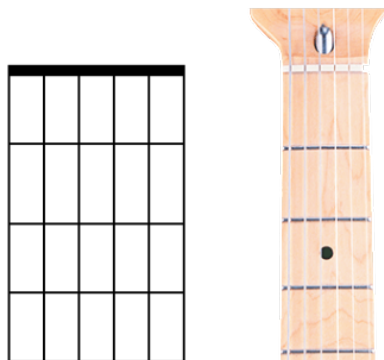


Figura 8 – Comparação entre diagrama de acordes e braço de uma guitarra

A partir da Figura 8, são destacados no diagrama:

- A linha superior horizontal é chamada de **pestanda**, parte de instrumento por onde as cordas são passadas. Também funciona como se fosse o traste zero do instrumento;
- As demais linhas horizontais são denominadas **trastes**. Divisões que estabelecem as posições, também conhecido como casas, no braço do violão;
- As **linhas verticais** são as cordas do instrumento que são representadas como “**E**, **B**, **G**, **D**, **A**, **E**” realizando a leitura da direita para a esquerda.

Uma característica importante são as distâncias entre as casas no braço do violão que se estabelecem em um intervalo de 01 semitom para cada posição. Assim, pressionando na corda **A** na 1^a posição, teremos então 01 semitom à frente, sendo **A[♯]** (lá sustenido) ou **B^b** (si bemol).

A seguir é mostrado o diagrama dos acordes **C** (dó maior) e **Cm** (dó menor) representados no braço do violão.

A partir do acorde montado sobre o diagrama mostrado na Figura 9, temos as seguintes informações adicionais:

⁸ Acordes de violão. Acessado em Março de 2017 de <<http://acordesdeviolao.com.br/acordes-de-violao-aula-definitiva/>>

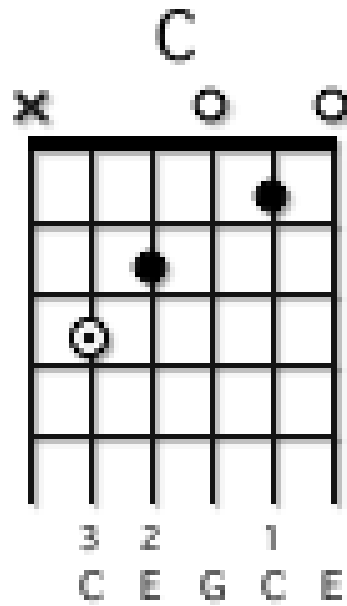


Figura 9 – Diagrama do C

- O “x” indica que a corda não deve ser tocada por não pertencer ao acorde;
- Os círculos que possuem o preenchimento total na cor branca e que estão na parte superior do diagrama indicam que as cordas referentes devem ser tocadas soltas;
- Os círculos que estão sobre às linhas verticais indicam quais cordas devem ser tocadas e em quais posições no braço do violão;
- O círculo que possui a cor branca e um ponto preto em seu centro indica que a posição atual é a nota principal do acorde, também chamada como tônica;
- Os números inteiros indicam quais dedos a partir do indicador devem ser utilizados;
- As letras na parte inferior indicam quais notas serão tocadas.

O mesmo procedimento ocorre para a formação do **Cm** (dó menor) mostrado na Figura 10. Porém, um número inteiro é mostrado ao lado esquerdo do diagrama, indicado em qual posição do braço é formado o acorde.

2.1.8 Cifras

A cifra é um tipo de notação que representa o nome dos acordes musicais. Esse tipo de notação pode ser visualizado acima das letras de uma composição musical por exemplo (Figura 11).

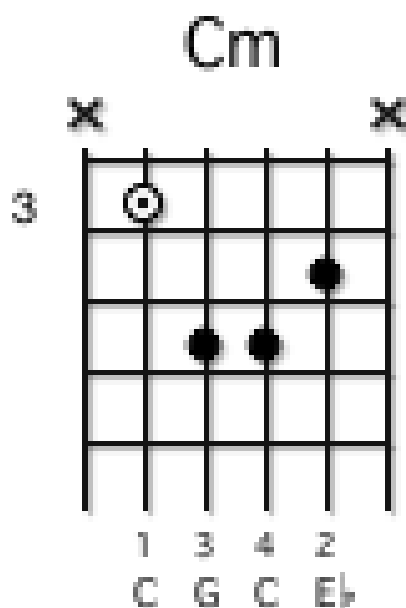


Figura 10 – Diagrama do Cm

G Em
 Old Pirates, yes, they rob I.
 C G Am
 Sold I to the merchant ships
 G Em C G Am
 Minutes after they took I from the bottom less pit.
 G Em
 But my hand was made strong
 C G Am
 By the hand of the Almighty.
 G Em C G D
 We forward in this generation triumphantly.

Figura 11 – Parte da cifra “Redemption Song - Bob Marley”

Assim sendo, é notória a presença de acordes maiores e menores acima das letras. Além dos acordes serem destacados para a facilitação da leitura, estes são posicionados em lugares específicos indicando o momento em que devem ser executados no instrumento ao se realizar o canto das estrofes.

2.2 Web Audio API

A inclusão de áudios na *Internet* contribuiu bastante para uma melhor interação dos usuários com as páginas *web*. Assim, plugins instaláveis como *Adobe Flash Player* (SOFTWARE, 2017) e *QuickTime* (INC, 2017) serviram de base para prover tal funcionalidade.

Com a chegada do *HTML5*, a mais recente evolução que define a linguagem

HTML para a construção de páginas *web* (NETWORK, 2016), surgiu o elemento *HTML5* `<audio>` que especifica uma maneira padronizada para incorporar áudios em uma página *web* (W3SCHOOLS, 1999c). Desta forma, não se faz mais necessário realizar instalações de plugins externos para as aplicações, porém o novo elemento de áudio trouxe consigo muitas limitações o que não o tornava poderoso o bastante para gerenciar as funções de áudio complexas. Assim, uma nova solução era preciso para atender às aplicações mais sofisticadas (SMUS, 2011).

Assim sendo, tem-se a *Web Audio API* (W3C, 2017), uma poderosa ferramenta de alto desempenho de manipulação de áudio nos navegadores através da linguagem de programação *JavaScript*. Todas as operações de áudio com a *Web Audio API* são gerenciadas dentro de um contexto de áudio, sendo denominada de *AudioContext*, bem como as operações são realizadas em nós de áudio, ou *audio nodes* (HOVANESYAN, 2017).

A arquitetura do fluxo da ferramenta é mostrada na Figura 12



Figura 12 – Fluxo de trabalho da *Web Audio API*

Conforme visto na imagem acima, todo o sistema de conexão entre os elementos estabelece um roteamento modular, permitindo a flexibilidade em criar funções de áudios complexas com efeitos dinâmicos. Logo, um fluxo de trabalho simples com a *Web Audio* pode ser descrito através dos itens abaixo:

1. Criar um contexto de áudio;
2. Criar fontes de áudio dentro do contexto como oscilador, *stream*, *samples*;
3. Criar efeitos de áudio presentes nos nós como *reverb*, compressor, *delay*;
4. Conectar o destino final do áudio como alto-falantes do sistema por exemplo;
5. Conectar as fontes de áudio (*source*) com os efeitos (*filter nodes*) e com o destino (*destination*).

Uma simples implementação para se tocar um áudio é mostrado na Figura 13.

Conforme visto na implementação na Figura 13, a variável *context* contém a instância do objeto *AudioContext*. Na função *playSound()*, a variável *source* é criada para poder receber de *dogBarkingBuffer* a pequena porção de áudio armazenada na memória e ser conectada ao seu destino, ou seja, alto-falante por exemplo, com o método *connect*. Por fim, o método *start()* faz com que o áudio seja tocado.

EXAMPLE 1

```
var context = new AudioContext();

function playSound() {
  var source = context.createBufferSource();
  source.buffer = dogBarkingBuffer;
  source.connect(context.destination);
  source.start(0);
}
```

Figura 13 – Implementação para tocar um áudio com a ferramenta *Web Audio API*

2.2.1 Bibliotecas de áudio

A chegada da *Web Audio API* mudou completamente o paradigma tradicional da forma se trabalhar com áudios na *Internet*. Agora, desenvolvedores interessados no assunto relacionado a áudios na internet possuem uma maior liberdade para desenvolver e gerenciar suas aplicações produzindo uma maior interação para os usuários, o que antes era pouco viável e bastante limitado simplesmente através dos plugins externos ou com o surgimento do *HTML5*.

Portanto, com a evolução da *Web Audio*, bibliotecas de áudio foram sendo desenvolvidas com base na *API* visando atender aos mais variados tipos de necessidades e proporcionando uma melhor experiência para os usuários.

A seguir, algumas bibliotecas serão abordadas descrevendo as suas principais funções bem como a utilização de uma delas para a construção deste projeto.

2.2.2 Theresa's Sound World

Baseado na *API* do *Web Audio*, o *tsw*, como é comumente chamado através de suas funções, utiliza o sistema de roteamento modular da *Web Audio* para criar áudios sofisticados e instrumentos de uma maneira rápida e amigável⁹.

Seguindo a analogia da conexão entre pedais de efeito de uma guitarra em que cada pedal modifica o som e sendo transmitido para outro pedal de efeito, o mesmo ocorre com a utilização do *tsw* com a função *connect()* (Figura 14).

```
1 // Connect the oscillator to a gain node, then connect the gain node to the speakers.
2 tsw.connect(osc, gainNode, tsw.speakers);
3 // Start the oscillator.
4 osc.start();
```

Figura 14 – Simples implementação do *TSW*

⁹ Theresa's Sound World. Acessado em Março de 2017 de <<https://github.com/stuartmemo/theresas-sound-world>>

Os pedais de efeito podem ser combinados sendo conectados uns aos outros para fazer sons diferentes. O mesmo ocorre na implementação da imagem da Figura 14, em que o *oscillator* é conectado ao *gainNode*, pedal de volume por exemplo, e por fim ao *speaker*, semelhante ao amplificador.

Atualmente, o código fonte do *tsw* está na versão 0.1.15 com menos de 100 kilobyte de tamanho.

2.2.3 Pedalboard.js

Pedalboard.js é um *framework JavaScript* de código aberto para o desenvolvimento de efeitos de áudio voltados para o instrumento de guitarra ¹⁰. Estando na versão 0.5.1, a aplicação é construída sob os conceitos dos efeitos proveniente dos pedais para o instrumento citado.

O *framework* possui classes que são modeladas a partir dos reais componentes do pedal de efeito de um instrumento real e do *pedal board*, plataforma que agrupa vários pedais de efeito. Portanto, a aplicação permite que o usuário configure os componentes da mesma forma que se configura um pedal.

Os principais componentes do *framework* são:

- *Stage*: onde toda a aplicação reside;
- *Board*: análogo ao pedal board, é onde residem os pedais;
- *Box*: é uma classe abstrata para todos os pedais que são instanciados.

Os componentes apresentados são mostrados na Figura 15.

2.2.4 Tone.js

Tone.js é um *framework* que tem a sua implementação baseada na *Web Audio API*, provendo recursos avançados, sintetizadores, efeitos e abstrações musicais intuitivas ¹¹. Atualmente, está na versão release r9 e contém um “*Hello World*” (Figura 16), passos iniciais na programação, em seu repositório no *GitHub*, um serviço de *web hosting* compartilhado para projetos que usam o controle de versionamento *Git*.

A escolha desta aplicação para a construção deste projeto levou em consideração a possibilidade de se alterar a tonalidade de um arquivo de áudio através da função *triggerAttackRelease()* do *Tone.js* de uma maneira mais objetiva e direta.

¹⁰ Pedalboard.js. Acessado em Março de 2017 de <<https://github.com/dashersw/pedalboard.js>>

¹¹ Tone.js. Acessado em Março de 2017 de <<https://github.com/Tonejs/Tone.js>>

```
1 // initialize the stage and get the context
2 var stage = new pb.Stage();
3 var ctx = stage.getContext();
4
5 // initialize the board and pedals
6 var board = new pb.Board(ctx);
7 var od = new pb.stomp.Overdrive(ctx);
8 var reverb = new pb.stomp.Reverb(ctx);
9 var vol = new pb.stomp.Volume(ctx);
10
11 // add pedals to board
12 board.addPedals([od, reverb]);
13 board.addPedalsAt(1, vol);
14
15 // tweak pedal settings
16 od.setDrive(0.7);
17 od.setLevel(0.7);
18 reverb.setLevel(0.3);
19 vol.setLevel(0.2);
20
21 // set the board on stage and start playing!
22 stage.setBoard(board);
```

Figura 15 – Utilização do `pedalboard.js`

```
//create a synth and connect it to the master output (your speakers)
var synth = new Tone.Synth().toMaster();

//play a middle 'C' for the duration of an 8th note
synth.triggerAttackRelease("C4", "8n");
```

Figura 16 – Utilização do `Tone.js`

2.3 Trabalhos correlatos

Foram selecionados dois trabalhos que possuem relações com as funcionalidades deste projeto: GarageBand (INC, 2017) e Decifra-me.

2.3.1 GarageBand

O *GarageBand* é uma aplicação disponível para o sistema operacional *MacOS* e *iOS*, ambos da *Apple*, que tem como objetivo possibilitar a criação de músicas. O aplicativo abrange tanto o público experiente no universo da música quanto leigos, trazendo consigo uma interface de usuário bastante simples de ser utilizada (INC, 2017). Com diversos instrumentos musicais incluídos como guitarra, violão, teclado e bateria, o app transforma qualquer usuário em um editor musical com os instrumentos virtuais. Uma das funcionalidades que se relacionam com este projeto é o Smart Guitar, permitindo o usuário executar os acordes com apenas um toque (Figura 17). Também existe a possibilidade de o usuário conectar o real instrumento como o violão por exemplo, para realizar as gravações bem como incluir os acordes musicais diretamente no aplicativo.

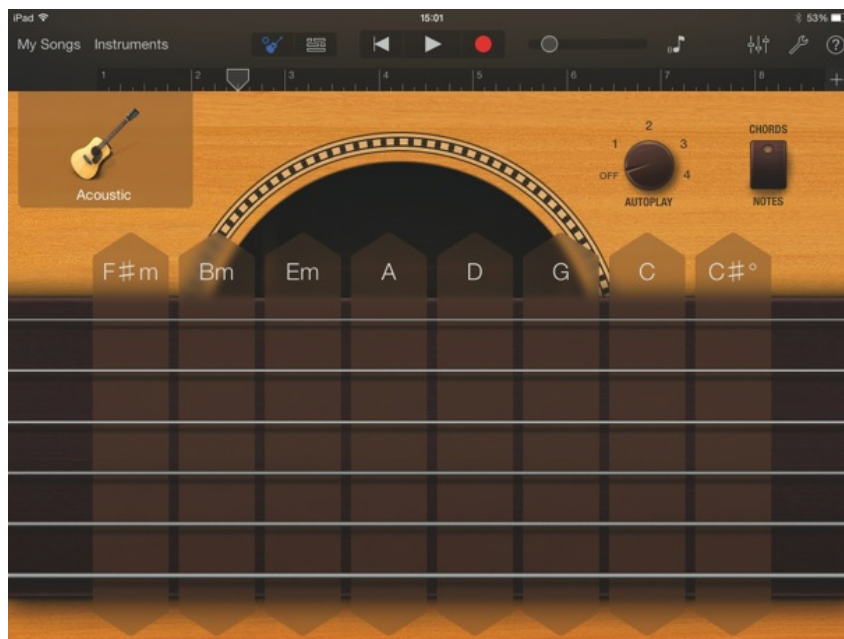


Figura 17 – Interface do GarageBand

2.3.2 Decifra-me

O Decifra-me é uma aplicação web utilizada no navegador voltada para *desktop* e dispositivos móveis como *tablet* e *smartphone* que faz uma listagem de cifras musicais provenientes do site de cifras Cifraclub.com.br. O objetivo da aplicação é saber quais acordes os usuários sabem ou não, permitindo ensiná-los através das cifras com os acordes ou músicas mencionados. Logo, caso o usuário saiba os acordes musicais maiores “C, D, E” por exemplo, o Decifra-me se encarrega de mostrar ao usuário as cifras que compõem os acordes que foram informados¹² (Figura 18).

¹² Decifra-me. Acessado em Março de 2017 de <<http://ciframe.herokuapp.com/#/>>



The image shows a screenshot of the Decifra-me website. On the left, there is a purple panel with the logo "Decifra-me" and the text "Aponte os acordes que você já domina". Below this is a grid of 24 colored hexagons representing chords: C, C#, Cm, C#m, C7, C#7, D, D#, Dm, D#m, D7, D#7, E, Em, E7, F, F#, Fm, F#m, F7, F#7, G, G#, Gm, G#m, G7, G#7, A, A#, Am, A#m, A7, A#7, B, Bm, B7. On the right, the text "Encontrei essas músicas" is followed by a dropdown menu set to "-- Gênero --" and four selected chord filters: C, Am, G, and F. Below this is a grid of nine music recommendations, each with a lock icon, title, artist, and genre tags.

Quando o Mundo Cai Ao Meu Redor	Homem Primata	O Que Eu Bebi
Livres Para Adorar	Titãs	Clarice Falcão
Gospel/Religioso	Pop Rock	Indie
Estamos Todos Bêbados	Love Me	Acabou
Matanza	Justin Bieber	Projota
Hardcore	Pop	Hip Hop/Rep
Nelore Valente	Mel Nesse Trem	Me Amar Amanhã
Let's Build a Planet	Clay A. Knight	Mathias & Kayne

Figura 18 – Listagem de cifras no Decifra-me

3 Desenvolvimento do aplicativo

3.1 Objetivo

Como descrito anteriormente em outras seções, o *Chordix*¹ tem como função principal prover ao usuário um auto acompanhamento vocal a partir de uma cifra musical através da execução dos acordes utilizando botões.

3.2 Funcionamento

Com o objetivo de atingir o público proveniente das plataformas *desktop* e dispositivos móveis, onde o número de usuários já ultrapassa o de computadores pessoais (GOOGLE, 2015), identificou-se a necessidade do desenvolvimento do projeto ser multi-plataforma. Logo, a disposição das telas são adaptadas de acordo com o tamanho e os recursos dos dispositivos em que segundo (MARCOTTE, 2010), *Web Design* responsivo, técnica de estruturação para a adaptação dos navegadores *web*, se relaciona às necessidades dos usuários e às plataformas que eles usam.

A utilização da aplicação pode ser observada na Figura 19, em que é disposto ao usuário uma tela contendo uma caixa de entrada para a inserção da *URL* da cifra proveniente do site Cifraclub.com.br, *website* contendo cifras e tablaturas para instrumentos, a qual o usuário deseja executar.

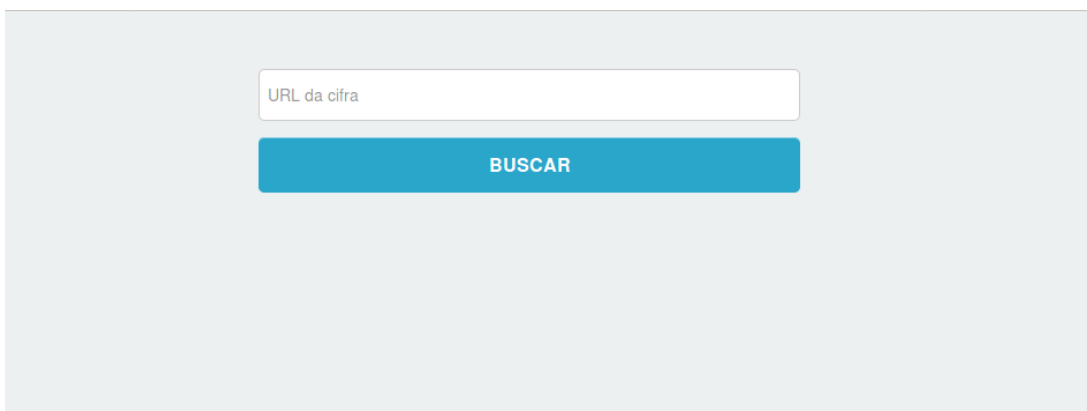


Figura 19 – Tela inicial

Após a inserção da *URL*, ao clicar no botão Buscar é mostrada uma segunda tela contendo a cifra requisitada conforme é observado na Figura 20.

¹ Chordix. Acessado em Março de 2017 de <<https://github.com/terciodejesus/chordix>>

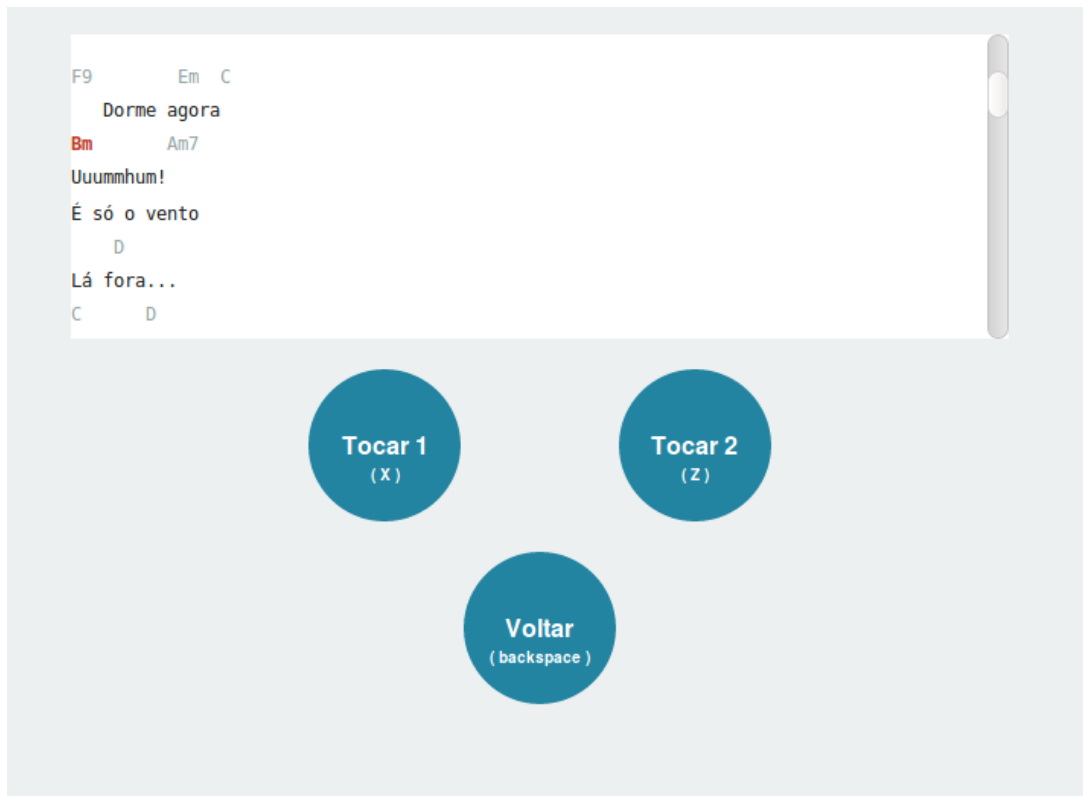


Figura 20 – Cifra carregada

Nesta segunda visualização, observa-se a presença de botões disponíveis ao usuário contendo ações que serão executadas na cifra representada, sendo a interação de forma sequencial na ordem em que os acordes são apresentados.

A interação ocorre da seguinte forma:

- Quando uma ação é feita no botão “Tocar 1” ou pressionando a letra Z no teclado, o próximo acorde é tocado e nele é permanecido, sendo permitido tocá-lo enquanto se pressiona o botão “Tocar 1” ou a letra Z no teclado;
- Quando uma ação é feita no botão “Tocar 2” ou pressionando a letra X no teclado, o próximo acorde é tocado e nele é permanecido, sendo permitido tocá-lo enquanto se pressiona o botão “Tocar 2” ou a letra X no teclado;
- Quando uma ação é feita no botão Voltar ou pressionando a tecla *backspace* do teclado, o acorde anterior é tocado e assim sucessivamente enquanto este botão ou a tecla *backspace* forem pressionados.

A presença de dois botões que possuem o mesmo efeito de executarem sempre o próximo acorde e nele permanecer garante que o usuário possa tocar continuamente o mesmo acorde pelo tempo que achar necessário, como em casos em que um acorde é tocado

por mais tempo. Assim, os acordes serão trocados à medida que os botões “Tocar 1” e “Tocar 2” forem tocados de forma alternada.

Visando garantir uma boa qualidade na experiência do usuário, percebe-se na Figura 20 que o acorde que está sendo tocado é destacado no corpo da cifra, fazendo com que a aplicação não atrapalhe durante as execuções.

3.2.1 Desenvolvimento

Serão apresentados nesta seção os detalhes da implementação do protótipo. Na seção 3.2.2 são descritas as tecnologias envolvidas na criação deste projeto. Na seção 3.2.3 e 3.2.4 são descritas as técnicas que foram implementadas e ferramentas utilizadas para o funcionamento da aplicação bem como os processos de identificação e extração dos acordes. Na seção 3.2.4 são descritos os processos para a execução dos acordes a partir da cifra musical.

3.2.2 Tecnologias Utilizadas

Pesquisas e estudos de tecnologias voltadas para o desenvolvimento web foram realizadas com o objetivo de facilitar o desenvolvimento da aplicação *Chordix*. Assim sendo, abaixo são listadas as aplicações existentes neste projeto:

1. **HTML5**: Linguagem de estrutura e apresentação dos conteúdos da aplicação *Chordix*;
2. **jQuery**: Biblioteca de desenvolvimento para a linguagem *JavaScript*, responsável por realizar a extração e identificação do conteúdo da cifra;
3. **Foundation**: *Framework* para o desenvolvimento de estilização e adaptação do *layout* para as diversas plataformas;
4. **Ruby on Rails**: *Framework* de desenvolvimento *web* para a implementação do *Web Service*;
5. **Heroku**: Plataforma de aplicações responsável por hospedar o *Web Service*;
6. **GitHub**: Plataforma de hospedagem de código para controle e versão da aplicação *Chordix* e para o *Web Service*.

3.2.3 Web Service

Para a obtenção da cifra do site do Cifraclub.com.br, foi implementado um *Web Service*², na qual é uma solução utilizada na integração e comunicação entre aplicações diferentes.

Logo, ao solicitar a cifra, o *Chordix* requisita a página *HTML* ao *Web Service* enviando a *URL* como um dado empacotado por meio de uma requisição *HTTP*, protocolo da camada de aplicação da *Internet* (ROSS KEITH W.; KUROSE, 2013). Portanto, o serviço criado torna-se o solicitante do recurso ao site do Cifraclub.com.br, que devolve através de uma requisição *HTTP* como resposta para o serviço e para a aplicação *Chordix*, estabelecendo assim um modelo cliente-servidor entre as aplicações (ROSS KEITH W.; KUROSE, 2013) conforme pode-se observar na Figura 21.

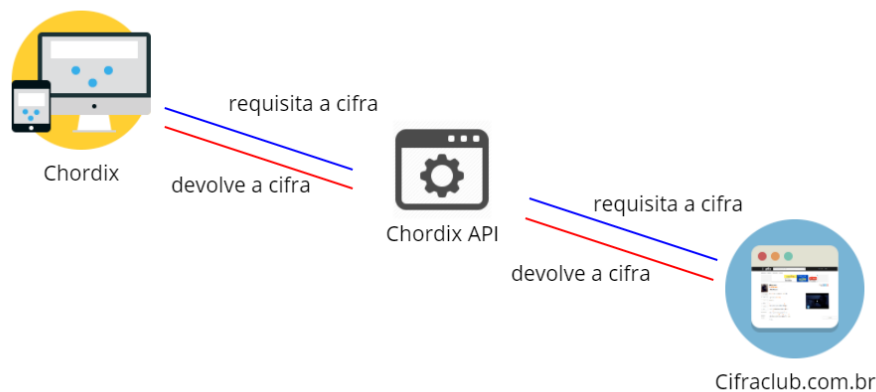


Figura 21 – Simples modelo cliente-servidor do Chordix

3.2.4 Extração e Identificação

Após o recebimento da página *HTML* proveniente do *Web Service* como citado na seção anterior, dois tipos de extrações e identificações são utilizadas para a obtenção dos conteúdos do corpo da cifra utilizando uma técnica de *web scraping*, método que consiste na extração de conteúdos de uma página *web*. São elas:

1. **Extração dos parágrafos:** seção que contivesse a classe (W3SCHOOLS, 1999a), seletor que identifica elementos *HTML*, proveniente da página *HTML* requisitada;
2. **Extração dos acordes:** todos os acordes presentes nos parágrafos que possuíssem o elemento ``, ganhando destaque com o efeito de negrito (W3SCHOOLS, 1999b).

² Chordix API. Acessado em Março de 2017 de <<https://github.com/terciodejesus/chordix-api>>

Após a extração dos acordes, identificou-se que os acordes provenientes do *Cifraclub* eram organizados em um *array* em *JavaScript*, uma espécie de lista na computação onde os dados são enfileirados, com informações dos nomes e formatos em que os acordes eram apresentados.

A estrutura é mostrada na Figura 22.

```
1 [
2   {
3     "chord": "Am7",
4     "alt": "Am7",
5     "guitar": [
6       "X 12 14 12 13 12",
7       "X 0 2 0 1 0",
8       "X 0 2 0 1 3",
9       "5 3 5 5 X X"
10    ],
11    "keyboard" : [
12      "10 1 5 8"
13    ],
14    "ukelele" : [
15      "0 0 0 0",
16      "2 4 3 3"
17    ]
18  }
19 ]
```

Figura 22 – Estrutura do acorde

Observando então a Figura 22, tem-se que nas linhas 3 e 4 é descrito o nome do acorde. O código fonte da cifra em questão não faz distinção entre o “*chord*” e “*alt*” por terem nomes diferentes mas, com o mesmo valor. Na linha 5, é importante notar a descrição de qual instrumento está sendo referenciado, visto que outras formações de acordes aparecem para outros instrumentos, como é mostrado a partir das linhas 11 e 14. Assim, todas as informações contidas neste projeto são referentes a um instrumento de violão de 06 cordas. Entre as linhas 6 e 9, são mostradas em formato de *string*, um conjunto de caracteres, as várias formas em que o acorde pode ser representado. Neste ponto, é visto que cada *string* possui 06 informações, sendo a mesma quantidade de cordas existentes em um violão de 06 cordas. Logo, realizando a leitura da esquerda para a direita dos conjuntos de caracteres, verifica-se que cada informação relaciona-se com as cordas do violão, sendo elas **E**, **A**, **D**, **G**, **B** e **E** respectivamente, seguindo na direção da corda de cima para a corda de baixo do instrumento.

A presença da letra **X** indica que a corda referente à essa posição não deve ser tocada e não faz parte do acorde.

A existência de números inteiros entre as informações contidas no conjunto de caracteres indica em qual posição, comumente chamada de **casa** no violão, a corda deve ser pressionada no instrumento. Assim, o acorde **Am7** sendo representado pela sequência de caracteres “**5 3 5 5 X X**” informa o seguinte:

1. A corda **E** é pressionada na posição **5** do violão;
2. A corda **A** é pressionada na posição **3** do violão;
3. A corda **D** é pressionada na posição **5** do violão;
4. A corda **G** é pressionada na posição **5** do violão;
5. A corda **B** não é tocada;
6. A corda **E** não é tocada.

3.2.5 Execução dos acordes

Para emitir o som dos acordes, são incluídos 06 arquivos de áudio referentes ao som emitido de cada corda solta de um violão em formato *wav*, um formato padrão de arquivo de áudio para o armazenamento de áudio em computadores, e atribuídas a 06 variáveis em *JavaScript* através da biblioteca ***Tone.js*** no momento em que a aplicação é iniciada. À medida que vão sendo identificados ao realizar ações tanto no botão “Tocar 1” quanto no botão “Tocar 2” visto na seção 3.2, a função *triggerAttack()* do ***Tone.js*** é invocada em cada variável criada, informada no início desta seção, passando como parâmetro para a função o número inteiro presente no vetor, que corresponde à posição da casa no violão. Assim, a sequência “**5 3 5 5 X X**” será executada da seguinte forma:

1. **e1.triggerAttack(5);**
2. **a.triggerAttack(3);**
3. **d.triggerAttack(5);**
4. **g.triggerAttack(5).**

As variáveis **b** e **e2** não serão tocadas por conter o **X** na sequência de caracteres referentes ao acorde, conforme explicado na seção anterior.

A informação do valor inteiro passado como parâmetro para o ***Tone.js***, refere-se à quantidade de semitons que serão aplicados ao som da corda original previamente carregado no momento em que a aplicação é iniciada. Portanto, quando a função é invocada no Item 1, são avançados 05 semitons. Logo, analogamente ao contexto de um violão, a corda **E1** é pressionada na posição **5** em seu braço.

O mesmo procedimento de execução das outras cordas ocorrem nos demais itens.

4 Avaliação

Para o processo inicial de desenvolvimento do *Chordix*, diversas pesquisas foram feitas em relação à utilização de plugins de áudio. Características como qualidade na documentação de uso dos plugins e na aproximação dos sons emitidos foram pontos cruciais para o seu desenvolvimento. Logo, o plugin *Tone.js* foi utilizado na implementação por proporcionar um maior controle na manipulação de áudios das cordas soltas de um violão previamente gravadas e a fácil alteração nas frequências dos sons podendo alterar suas tonalidades.

Realizar a identificação da seção do conteúdo das estrofes juntamente com os acordes musicais se deu de forma satisfatória para a versão *desktop* através da técnica de *web scraping* em conjunto da função *find()* do *jQuery*, biblioteca para a linguagem *JavaScript* para simplificar o desenvolvimento, após a obtenção da página *HTML* informada pela *URL*. No entanto, problemas foram encontrados ao requisitar uma cifra com a identificação da letra **m** na *URL*, responsável por renderizar um layout diferente para um dispositivo móvel, contendo uma estrutura diferente. Porém, o modo de visualização da cifra pode ser alternado para “**versão clássica**” que fica disposto ao fundo da página, gerando uma *URL* que atende à estrutura da aplicação solucionando o problema.

Para a execução dos acordes, os sons emitidos aconteceram de forma esperada em que as 06 variáveis contendo referências às cordas são executadas ao mesmo tempo no momento em que o usuário clica ou pressiona o botão de execução. Porém, um problema foi encontrado ao se realizar múltiplas ações de execuções dos acordes em pequenas frações de segundos, resultando em uma mistura de sons, mas não sendo uma barreira para o andamento da cifra.

O tempo de resposta ao realizar execuções dos acordes é de forma instantânea na plataforma *desktop* através dos navegadores *Google Chrome* e *Mozilla Firefox*, provendo boas interações para o usuário por não conter atrasos. Porém, testes foram realizados nos dispositivos móveis em ambos os *browsers* de navegação e o tempo de resposta não aconteceu de forma instantânea em alguns dispositivos. No *Chrome*, aparelhos com o sistema operacional *Android* não tão modernos como *MOTO G 1ª* geração da Motorola e *LG K4* apresentaram atrasos nas execuções, trazendo consigo uma maior dificuldade na sua execução mas, não inviabilizando o processo. No entanto, testes feitos no Firefox apresentaram ruídos ao se executar os acordes.

Análises foram feitas no *smartphone MOTO G4 Plus* da Motorola que apresentaram o mesmo comportamento da plataforma *desktop*, provendo bons desempenhos juntamente com a sincronia das execuções e sensibilidade ao realizar ações em uma tela *touchscreen*.

O *Galaxy Tab 2* que também pertence à linha *Android* não teve um desempenho esperado por não ter suporte ao plugin. Por fim, dispositivos móveis com o sistema operacional *iOS* apresentaram falhas para o funcionamento da *Web Audio API*¹ juntamente com o plugin *Tone.js*, o que pode ser uma consequência da mais nova atualização do sistema para a versão 10².

¹ Web Audio API Test. Acessado em Março de 2017 de <https://www.sharphosting.uk/ipad_test.html#>

² Web Audio Api Broken on IOS 10.1.1 ipad. Acessado em Março de 2017 de <<http://wla.berkeley.edu/~cs61a/fall/lectures/communication.html>>

5 Conclusão

Foi abordado neste trabalho os processos de construção de um aplicativo para o auto-acompanhamento musical para ambientes multiplataforma voltados para usuários no âmbito da instrumentação vocal.

Tendo em vista os resultados obtidos através de testes e experimentos realizados nas etapas de construção, pode-se concluir que este projeto alcançou o objetivo de um modo satisfatório, permitindo que usuários executem acordes musicais a partir de cifras *HTML*.

A principal limitação da aplicação se encontra no fato de que a latência é alta em dispositivos móveis clássicos com o sistema operacional *Android*, não proporcionando uma boa experiência de usuário ou a falta de compatibilidade dos *browsers* com o plugin *Tone.js*, inviabilizando o processo. Para uma melhor interação, é necessário a utilização em um ambiente *desktop* ou a obtenção de um *smartphone* ou *tablet* que contenha um desempenho melhor implicando em custos maiores para a aquisição. Com dispositivos móveis da *Apple*, dificuldades foram encontradas para a execução dos acordes que inviabilizaram o processo decorrente da atualização do sistema para a versão 10.

As manutenções corretivas e evolutivas devem ser realizadas de maneira periódica para o processo de extração e identificação das cifras, uma vez que alterações na estrutura *HTML* do site *Cifraclub.com.br* sejam realizadas pelos proprietários.

Para trabalhos futuros deste projeto com o intuito de avaliar a usabilidade da aplicação, análises qualitativas e quantitativas deverão ser coletadas através de testes de experimentos e questionários com usuários de diferentes perfis musicais com o objetivo de serem avaliados ao realizar a instrumentação vocal enquanto tocam os acordes a partir das cifras.

5.1 Trabalhos Futuros

- A Utilizar o *framework* de desenvolvimento multiplataforma *Apache Cordova*¹ para criar e testar a aplicação com funcionalidades específicas ao ambiente desenvolvido como *Android* ou *iOS*;
- B Permitir a utilização de cifras *HTML* previamente contidas no armazenamento dos dispositivos móveis, evitando acessos ao site do *Cifraclub* a partir da aplicação criada

¹ Apache Cordova, Mobile apps with HTML, CSS & JS. Acessado em Março de 2017 de <<https://cordova.apache.org/>>

com o *Apache Cordova*;

- C Permitir a extração e identificação da cifra a partir da *URL* voltada para dispositivo móvel;
- D Incluir funcionalidade de alteração de tonalidade a partir da cifra original;
- E Prover funcionalidade para listagem de cifras e consultas a partir dos títulos;
- F Realizar extrações de outros sites como cifras.com.br.

Referências

- ABDOUNUR, O. J. *Matemática e Música: o pensamento analógico na construção de significados*. [S.l.: s.n.], 2006. ISBN 9788586303524. Citado na página 27.
- ALVARENGA, G. G. Ferramenta para criação de composições musicais para android. 2013. Citado na página 23.
- BENWARD, B. *Music in Theory & Practice*. [S.l.: s.n.], 2003. ISBN 978-0074121153. Citado na página 29.
- GOOGLE. *Google Mobile Day*. 2015. <<https://www.thinkwithgoogle.com/intl/pt-br/articles/por-que-mobile.html>>. Acessado em Março de 2017. Citado na página 41.
- HOVANESYAN, G. *Getting Started with Web Audio API*. 2017. <<https://css-tricks.com/introduction-web-audio-api/>>. Acessado em Março de 2017. Citado na página 34.
- INC, A. *QuickTime*. 2017. <https://support.apple.com/kb/dl837?locale=pt_BR>. Acessado em Março de 2017. Citado 2 vezes nas páginas 33 e 37.
- KROGER, P. *Music for Geeks and Nerds*. [S.l.]: CreateSpace Independent Publishing Platform, 2012. ISBN 1478345381. Citado 2 vezes nas páginas 26 e 28.
- MARCOTTE, E. Responsive web design. 2010. Citado na página 41.
- NETWORK, M. D. *HTML5*. 2016. <<https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>>. Acessado em Março de 2017. Citado na página 34.
- PATRICIO, E. L. B. *criação e questões dos instrumentos musicais digitais*. 2012. <<http://www.semeiosis.com.br/wp-content/uploads/2012/06/PATRICIO-Eduardo-Luis-Brito.-Criacao-e-questoes-dos-instrumentos-musicais-digitais.pdf>>. Acessado em Março de 2017. Citado na página 23.
- PRIORE, I. O desenvolvimento da teoria musical como disciplina independente: princípio, conflitos e novos caminhos. *OPUS - Revista Eletrônica da ANPPOM*, v. 19, n. 1, p. 9–26, 2015. ISSN 1517-7017. Citado na página 25.
- ROSS KEITH W.; KUROSE, J. *Redes de Computadores e A Internet - Uma Abordagem Top-Down*. [S.l.: s.n.], 2013. ISBN 9788581436777. Citado na página 44.
- SANTO, P. R. da F. *Sistema Internacional de Unidades*. 2006. <http://www.inmetro.gov.br/inovacao/publicacoes/si_versao_final.pdf>. Acessado em Março de 2017. Citado na página 26.
- SFETCU, N. *The Music Sound*. [S.l.]: Nicolae Sfetcu, 2014. ISBN 9990044519112. Citado na página 25.
- SMUS, B. *Getting Started with Web Audio API*. 2011. <<https://www.html5rocks.com/en/tutorials/webaudio/intro/>>. Acessado em Março de 2017. Citado na página 34.
- SOARES, L. C. *Pitágoras e a Música*. 2006. <<http://www.ghtc.usp.br/server/Sites-HF/Lucas-Soares/Home.html>>. Acessado em Março de 2017. Citado na página 28.

- SOFTWARE, A. S. *Adobe Flash Player*. 2017. <<https://get.adobe.com/br/flashplayer//>>. Acessado em Março de 2017. Citado na página 33.
- W3C. *Web Audio API*. 2017. <<https://webaudio.github.io/web-audio-api/>>. Acessado em Março de 2017. Citado na página 34.
- W3SCHOOLS. *CSS Selector*. 1999. <https://www.w3schools.com/cssref/sel_class.asp>. Acessado em Março de 2017. Citado na página 44.
- W3SCHOOLS. *HTML Text Formatting*. 1999. <https://www.w3schools.com/html/html_formatting.asp>. Acessado em Março de 2017. Citado na página 44.
- W3SCHOOLS. *HTML5 Audio*. 1999. <https://www.w3schools.com/html/html5_audio.asp>. Acessado em Março de 2017. Citado na página 34.
- WEITGENANT, F. *My music: Aplicativo de composição musical para leigos*. 2014. Citado na página 23.