

Técnicas de Construção de Programas

Trabalho Prático – Fase 1

Semestre 2020/2

Professor Marcelo Soares Pimenta

Grupo: Jordi Pujol, Lucas Bombana, Rafael Petry e Victória Duarte

REQUISITOS

REQUISITOS FUNCIONAIS			
RF	Descrição	Prioridade	RF Relacionado
RF01	O software deve permitir transformar uma letra A ou a em som para a nota Lá.	A	-
RF02	O software deve permitir transformar uma letra B ou b em som para a nota Si.	A	-
RF03	O software deve permitir transformar uma letra C ou c em som para a nota Dó.	A	-
RF04	O software deve permitir transformar uma letra D ou d em som para a nota Ré.	A	-
RF05	O software deve permitir transformar uma letra E ou e em som para a nota Mi.	A	-
RF06	O software deve permitir transformar uma letra F ou f em som para a nota Fá.	A	-
RF07	O software deve permitir transformar uma letra G ou g em som para a nota Sol.	A	-
RF08	O software deve permitir transformar um caractere espaço em um silêncio ou pausa.	A	-
RF09	O software deve permitir aumentar volume para o dobro do volume atual ao receber um caractere + .	M	-
RF10	O software deve permitir diminuir volume para o	M	-

	volume default ao receber um caractere -.		
RF11	O software deve permitir repetir nota anterior ou pausa ao receber a vogal O ou o .	M	-
RF12	O software deve permitir repetir nota anterior ou pausa ao receber qualquer a vogal I ou i .	M	-
RF13	O software deve permitir repetir nota anterior ou pausa ao receber a vogal U ou u .	M	-
RF14	O software deve permitir aumentar uma oitava ao receber a letra T seguido do símbolo de adição + .	M	-
RF15	O software deve permitir diminuir uma oitava ao receber a vogal T seguido do símbolo de subtração - .	M	-
RF16	O software deve poder gerar uma nota aleatória de A até G.	M	RF17 RF18
RF17	O software deve permitir tocar uma nota aleatória (de A até G) ao receber um ponto de interrogação (?).	M	RF16
RF18	O software deve permitir tocar uma nota aleatória (de A até G) ao receber um ponto final (.)	M	RF16
RF19	O software deve gerenciar uma pilha com os possíveis instrumentos do programa.	M	RF20 RF21
RF20	O software deve permitir criar um ciclo com a pilha de instrumentos.	M	RF19 RF21
RF21	O software deve permitir trocar de instrumento ao receber um caractere NL (nova linha).	M	RF19 RF20
RF22	O software deve permitir aumentar o BPM em 50 unidades ao receber as letras BPM seguido do símbolo de adição + .	M	-
RF23	O software deve permitir diminuir o BPM em 50 unidades ao receber as letras BPM seguido do símbolo de subtração - .	M	-
RF24	O software deve continuar em seu estado atual ao receber um caractere com um caso não definido.	M	-

REQUISITOS NÃO FUNCIONAIS		
RNF	Descrição	Tipo de requisito
RNF01	Caso um erro aconteça, o usuário deve ser alertado de forma clara acerca do motivo do erro.	Confiabilidade / Facilidade de Uso
RNF02	O software deve rodar em máquinas com pelo menos 1Gb	Desempenho

	de memória RAM e capacidade de processamento de um processador Pentium I	
RNF03	O software deve dar respostas em no máximo 5 segundos.	Desempenho
RNF04	O software deve ser desenvolvido em até seis meses.	Entrega
RNF05	O software deve ser intuitivo.	Facilidade de uso
RNF06	O software deve ser treinado durante 1 dia.	Facilidade de Uso
RNF07	O software deve possuir um frame de ajuda por página.	Facilidade de Uso
RNF08	O software deve ser um executável.	Implementação
RNF09	O software deve ser desenvolvido em Java.	Implementação
RNF10	O software deve ser orientado a objetos.	Implementação
RNF11	O software deve funcionar em máquinas que utilizam a versão 8 ou superior do Java.	Portabilidade

REGRAS DE NEGÓCIO	
RN	Descrição
RN01	O software não deve possuir integração com mais do que três instrumentos.
RN02	O software não deve diferenciar letras maiúsculas ou minúsculas para a entrada de texto.
RN03	O software deve limitar o BPM em até 200 unidades.
RN04	O volume default do software deve ser de 20% da capacidade do sistema
RN05	O software deve limitar o volume em 100% da capacidade do sistema.
RN06	O software deve limitar texto de entrada com 256 caracteres.

OPERAÇÕES DO USUÁRIO		
OU	Operação	Alteração
OU01	Digitar um texto	O texto aparecerá no campo
OU02	Digitar nome do arquivo	O texto aparecerá no campo
OU03	Escolher instrumento	Ao clicar na aba de seleção (em um primeiro momento vazia), esta abrirá mostrando os instrumentos disponíveis. Quando o instrumento for selecionado, a aba encolhe e o nome do instrumento aparece na tag.

OU04	Gerar música	Ao clicar, a aba com o play da música aparece na tela
OU05	Iniciar música	A música é iniciada
OU06	Baixar música	A música é salva no computador

CLASSES

Enum NoteEnum	
Nota	Valor
Dó	0
Ré	2
Mi	4
Fá	5
Sol	7
Lá	9
Si	11

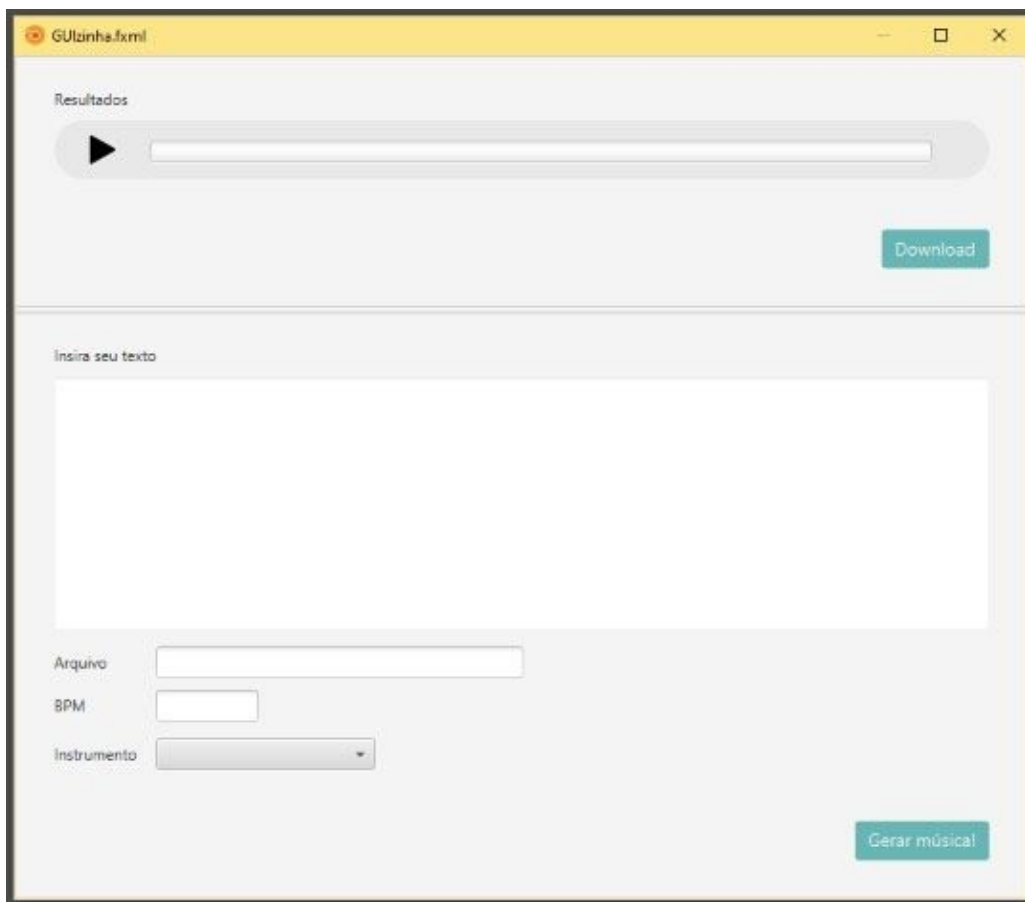
Randomizer
Métodos
randomize (range)

TextToMusic
Atributos
int current_octave → oitava do momento
int current_instrument → instrumento atual (primeiramente escolhido pelo tal do usuário)
int current_volume → volume atual da fucking música
int current_BPM → BPM atual (primeiramente escolhido pelo tal do usuário)
NoteEnum current_note → nota atual
Métodos
construtor (int current_instrument, int current_BPM)

```
convert (raw_text) { return text_musicado }
```

MusicPlayer
Atributos
Player player → objeto da biblioteca JFugue
var music → música a ser tocada
Métodos
construtor (music)
playMusic () → toca a música
saveMusic (filename) → salva a música

INTERFACE DE USUÁRIO



A interface criada busca ser o mais simples e intuitiva possível. Tentamos manter todas as funcionalidades disponíveis em um único local.

A GUI é dividida em duas áreas: a área de criação da música e a área de execução e download da música. Na primeira área, o usuário pode inserir seu texto, escolher o nome do arquivo para salvar a música e escolher o BPM e o instrumento inicial. A segunda região tem o player, onde o usuário poderá ouvir a música gerada, e um botão para fazer download da música.