

# INF01120 - TÉCNICAS DE CONSTRUÇÃO DE PROGRAMAS

## Trabalho Prático – Fase 1

Pedro Afonso Tremea Serpa – 275960

Bruno Longo Farina – 263412

Rafael Weber Becker - 301368

Luis Roberto Coelho - 314111

## Requisitos Funcionais

- Ter um caixa de input de texto na interface
- Possibilitar a entrada de um arquivo txt como input
- mapear um char para uma nota conforme definido
- aumentar o volume para o dobro do atual quando ler o char '+'
- ao ler “!” troca-se o instrumento para General MIDI #114 (Agogo)
- ao ler O,o,I,i,U ou u, troca-se o instrumento para o instrumento General MIDI #7 (Harpsichord)
- ao ler uma consoante não definida, se o caractere anterior era nota, repete-se a nota. Caso contrário, pausa.

- Se o caractere for um dígito, troca-se o instrumento General MIDI cujo número é igual ao valor do instrumento atual + valor do dígito
- ao ler "?" ou "." aumenta uma oitava. Se não puder aumentar, volta ao default
- ao ler '\n' deve trocar de instrumento para General MIDI #15 (Tubular Bells)
- ao ler ";" trocar instrumento para o instrumento General MIDI #76 (Pan Flute)
- ao ler "," trocar instrumento para o instrumento General MIDI #20 (Church Organ)
- ao ler qualquer outro caractere não definido, se caractere anterior era NOTA (A a G), repete nota; Caso contrário, Silêncio ou pausa

## Requisitos não Funcionais

- O software deve receber como entrada um texto ou um arquivo de texto
- O software deve exibir o volume atual, o ritmo atual e a escala atual, tendo como referência a nota A(lá)
- O software deve possuir valores default para as variáveis volume, ritmo e escala, sendo permitida a alteração das mesmas pelo usuário antes da execução do programa

- O software deve ser intuitivo
- O software deve ser um executável.
- O software será desenvolvido em Java.

## CLASSES

GUI
Atributos
<ul style="list-style-type: none"><li>- mintGreen: Color</li><li>- button_import: JButto</li><li>- button_export: JButton</li><li>- button_play: JButton</li><li>- button_mid: JButton</li><li>- textarea: JTextArea();</li><li>- scrollableTextArea: JScrollPane</li><li>- strToMusic: String</li></ul>
Métodos

+ main(): void  
+ GUI(): void  
+ actionPerformed(ActionEvent): void  
+ getCurrentNote(): String

## StringToPattern

### Atributos

### Métodos

+ convert(String): pattern

## PatternBuilder

### Atributos

- instrumentNumber: int
- octave: int
- volume: int
- stringPattern: StringBuilder

### Métodos

- + PatternBuilder(): void
- + getOutput(): Pattern
- + addNote(Character): boolean
- + doubleVolume(): void
- + changeInstrument(int): void
- + nextOctave(): void
- + addPause(): void

# Interface com o usuário

