



Trabalho de Técnicas de Construção de Programas

PARTE 1:

1.1) Requisitos:

1.1.1) Requisitos Funcionais

- Interpretador Python: Precisaremos do Interpretador Python para que seja possível ler o código que foi escrito e convertê-lo em bytecode (código de máquina).

- Debugger específico para Python: Permite a colocação de breakpoints para melhor visualização e entendimento do código, nos ajudando a entender o passo a passo do fluxo do código e garantindo a validação de trechos de código no software.

- Framework para teste do Software em Python: Necessário para testar as entradas e validar suas saídas. Ajuda e reforça a modularização do código, já que podemos testar cada módulo separadamente (respeitando o seguindo o “Modelo V” para testes)

- IDE com suporte à linguagem Python: Necessário para escrita e execução o software, conjuntamente com o Interpretador Python.

- GitHub: Repositório do código para auxiliar no desenvolvimento do software.

- Framework para Interface Python: Necessário para usabilidade, permitindo que o cliente utilize o software.

- Sistema Operacional Windows 7 (ou maior) e/ou Sistema Operacional Ubuntu 14.04 (ou maior): Necessário para o gerenciamento de recursos do programa e para fornecer suporte a entrada e saída do software.

1.1.2) Requisitos Não-Funcionais

1.1.3) Computador: Hardware necessário para realização e concretização do software.

- Monitor: Necessário para a visualização da execução do software.

- Teclado: Necessário para o gerenciamento de entrada no software.

- Mouse: Necessário para o gerenciamento de controle no software.

- Interface Gráfica do Usuário: Existem boas razões para que usemos uma GUI, um programa de computador com uma GUI bem planejada pode ser usada por qualquer usuário, independente de quão complexo o programa seja.

- Caixa de Som: Necessário para avaliação do conteúdo do software.

- Drivers de som atualizados: Necessário para avaliação do conteúdo do software, além de ser o total responsável pela saída do software.

1.2) Cronograma

23/10: Entrega da Fase 1: Entrega dos Requerimentos, definições de classes e módulos e interface.

09/11: Entrega da Fase 2: Entrega da Implementação dos módulos, arquivos de testes.

1.3) Projeto de definição de classes: Abaixo estarão listadas as classes, atributos, métodos e TAD's que serão possivelmente utilizados no trabalho final.

1.3.1) Classes:

- Classe de Tratamento de Texto: Responsável pelo recebimento do texto que o cliente digitar. Aqui ocorrerá a verificação do texto para certificar-se que será um texto permitido.
- Classe de Som: Aqui estará presente todos os sons que o enunciado do trabalho pede.
- Classe das Interfaces: Responsável pela criação e gerenciamento da interface de interação com o cliente.

1.3.2) Métodos:

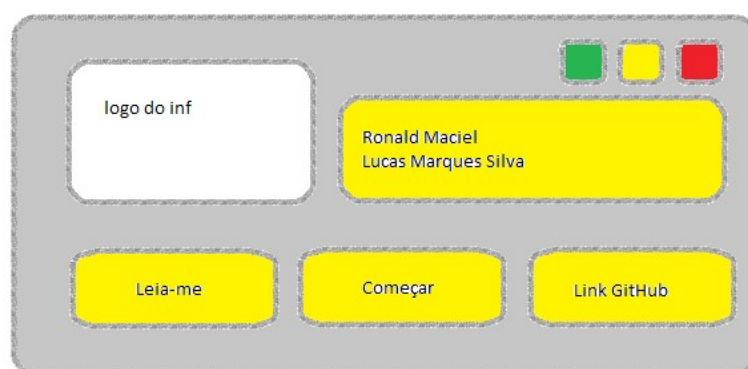
- Método de recebimento de texto: Salva o texto em uma variável.
- Método que verifica a validade do texto: Garante que o texto estará dentro do permitido, segundo o enunciado do trabalho.
- Método que mapeia o texto: Aqui será mapeado cada caractere que compõe o texto, gerenciando-os para suas devidas funções conforme os requisitos do trabalho.
- Método que gerencia as interfaces: Responsável pelo total funcionamento das interfaces – tais como: a tela da funcionalidade “Leia-me”.
- Método que gerencia o som: Responsável pelo aumento ou diminuição da intensidade sonora, conforme especificado no enunciado.
- Método de inicialização da música: Aqui será funcional com o botão presente na interface; permite que o cliente possa iniciar, em forma de música, o texto que foi digitado para o software.

1.3.3) Atributos:

- Atributo do volume: Guarda a configuração da intensidade da música.
- Atributo de texto: Guarda o texto que será revertido em forma de música.
- Atributo de BPM: Guarda a configuração de batidas por minuto da música.
- Atributo notas musicais: Guarda todas as notas musicais para serem mapeadas para os caracteres do texto.
- Atributo das oitavas: Guarda a configuração de uma oitava na música.
- Atributo de comprimento das interfaces: Guarda o comprimento das interfaces.
- Atributo de largura das interfaces: Guarda a largura das interfaces.

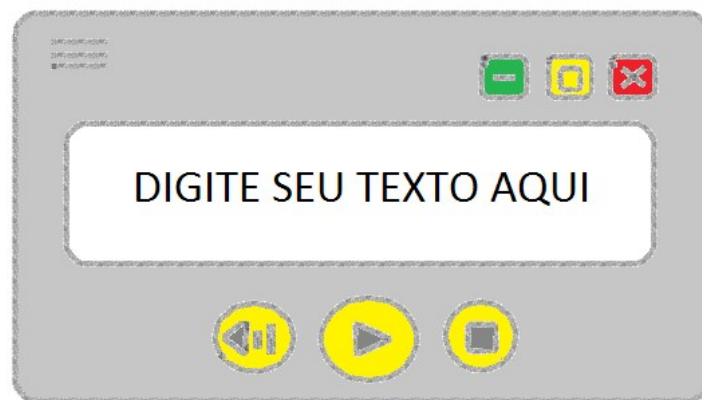
1.4) Projeto das Interfaces com o Usuário

INTERFACE INICIAL DO SOFTWARE:



- Leia-me: Descrição da utilização do software; arquivo-texto que funciona como manual do software.
- Começar: Atalho que redireciona o cliente para a janela principal para a utilização do software.
- Link GitHub: Atalho que redireciona o cliente para o site do GitHub, onde estão localizados os códigos-fonte utilizados no software.

INTERFACE SECUNDÁRIA DO SOFTWARE



- Caixa de texto: Permite ao cliente que escreva um texto (dentro do que for permitido) na caixa de texto para a leitura e, posteriormente, conversão desse texto em notas musicais.
- Botão de Play: Começa a tocar a música de acordo com a entrada que será digitada na caixa de texto presente acima.
- Botão de Pause: Para de tocar a música.
- Botão de volume: Altera a intensidade do volume.
- Botão com três riscos: Atalho que redireciona o cliente para a tela de interface inicial do software.

.... Fase 3...