

Desenvolvendo um Programa Registrador de Livros

Desenvolvedor: Anonmixs Amaro Afonso

Nome do Projeto: Registrador de livros

Público Alvo: Donos(as) de livraria, bibliotecas ou pessoas que queiram organizar seus livros em casa.

IDE: Spyder

Bibliotecas: PySimpleGUI

Plataformas Suportadas: GNU/Linux e Windows(7/10/11)

Sumário:

1- Objetivos	
1.1 Apresentação do projeto.....	3
2- Começo do Projeto	
2.1 Esquematizando a Interface Gráfica do Usuário(GUI).....	4
3- Visão Geral da Implementação em Python	
3.1 Produzindo o Código do Programa.....	6
4- Programa Finalizado	
4.1 Alguns Prints do Projeto Finalizado.....	8
5- Conclusão do Projeto	
5.1 Implementação do Projeto em uma livraria ou biblioteca de pequena escala.....	10

Capítulo 1

Objetivos

1.1 Apresentação do projeto

Ao longo desse relatório será mostrado o desenvolvimento de um programa multiplataforma que tem como objetivo facilitar o cadastramento de novos livros, assim como auxiliar na organização do inventário de uma livraria ou biblioteca.

Como plataforma de desenvolvimento será utilizado a IDE Spyder de código aberto e de uso gratuito, bem como a toolkit PySimpleGUI para o fornecimento de uma interface gráfica para o programa.

Capítulo 2

Começo do Projeto

2.1 Esquematizando a Interface Gráfica do Usuário(GUI)

Para que se tenha uma ideia de como deve ser a estrutura gráfica do programa final, será usada uma ferramenta de edição de imagens para desenhar o esquema gráfico do registrador de livros.

Como entradas do programa, ou as caixas onde o usuário digitará as informações de cadastro de cada livro novo, teremos:

- 1- Nome do livro
- 2- Autor
- 3- Tradutor
- 4- ISBN do livro
- 5- Editora
- 6- Sinopse do livro
- 7- Numero da edição e data
- 9- Data da entrada do livro na livraria/biblioteca
- 10- Preço sugerido(para livraria)
- 11- Estado do livro(Novo/Usado)
- 12- Se o livro for usado Breve descrição do estado de conservação do mesmo

Observa-se que tais entradas são apenas sugestões, sendo facilmente mudadas de acordo com a necessidade do cliente.

Utilizando o GIMP(programa de edição de imagens) tem-se a seguinte estrutura sugerida:

Registrador de livros
Livraria Novo Aprender

Dados do livro

Nome do Livro:

Autor:

Tradutor:

ISBN:

Editora:

Número da Edição e Data:

Data de entrada:

Preço de capa:

Sinopse:

Qual é o estado de conservação do livro:

☐ Novo ☐ Usado

Concluir

Se a opção "Usado" estiver marcada

Breve descrição do estado de conservação do livro:

Enviar

Se a opção "Mostrar resumo..." estiver marcada

Resumo

Nome do Livro: *****

Autor: *****

Tradutor: *****

ISBN: *****

.

.

Fechar o resumo

Janela que sempre se abrirá para que o usuário escolha se quer ou não ler um resumo do cadastro feito:

Mostrar resumo?

Sim

Não

Janela que se abrirá ao final de cada livro cadastrado:

Editar Cadastro

Salvar Cadastro e Sair

Salvar Cadastro e Cadastrar Outro

Cadastrar Outro Sem Salvar Atual

Finalizar Programa

Figura 1: Esquema sugerido para o programa

Agora pode-se partir para a parte da programação do “registrador de livros”, o qual será feita utilizando-se da linguagem Python.

Capítulo 3

Visão Geral da Implementação em Python

3.1 Produzindo o Código do Programa

Como a parte do designer e estrutura da interface gráfica já foram definidas, podemos passar para a parte do código. Relembrando será usado a IDE Spyder com a linguagem de programação Python.

A estrutura do código será da seguinte maneira:

Após cadastrar os dados se o usuário marcar a opção do livro como “Usado”, abrirá uma nova janela para que ele declare a condição atual do livro. Logo após surgirá uma janela que perguntará se o resumo do cadastro deve ser mostrado, após o resumo ser finalizado abrirá a janela para a escolha do próximo passo do programa(editar cadastro, salvar e sair, salvar e cadastrar outro livro, cadastrar outro livro sem salvar o atual e finalizar o programa sem salvar).

Ao final o usuário poderá escolher se os dados serão salvos no arquivo .txt(se o mesmo não existir no computador do usuário será criado). Se o cadastrante quiser fazer o cadastro de mais um livro, ele poderá clicar no botão “Cadastrar outro livro” e poderá refazer todo o processo novamente. É importante lembrar que cada cadastro ficará no mesmo arquivo .txt original para facilitar futuras consultas, se o usuário clicar em “Finalizar” o programa irá ser encerrado sem salvar o cadastro atual.

Para criar a interface gráfica será usado a ferramenta PySimpleGUI como comentado na introdução, cada janela implementada nessa ferramenta foi tratada como uma função que será chamada de acordo com o necessário.

```

3
4 # Janelas que serão executadas e Layouts
5 def janela_cadastro():
6     ps.theme('DarkBlue2')
7     fontt=20
8     layout = [
9         [ps.Text('Nome do Livro:\n', size=(14,1)), ps.Input(key='nomelivro', size=(60,1))],
10        [ps.Text('Autor:\n', size=(14,1)), ps.Input(key='autor')],
11        [ps.Text('Tradutor:\n', size=(14,1)), ps.Input(key='tradutor')],
12        [ps.Text('ISBN:\n', size=(14,1)), ps.Input(key='isbn')],
13        [ps.Text('Editora:\n', size=(14,1)), ps.Input(key='editora')],
14        [ps.Text('Edição(data):\n', size=(14,1)), ps.Input(key='edicao')],
15        [ps.Text('Data de Entrada:\n', size=(14,1)), ps.Input(key='datadeentrada')],
16        [ps.Text('Preço de capa:\n', size=(14,1)), ps.Input(key='preco')],
17        [ps.Text('Sinopse:\n', size=(14,1)), ps.Multiline(size=(48,10), key='textosinopse')],
18        [ps.Text('\n')],
19        [ps.Text('----- Qual é o estado de conservação do livro -----')],
20        [ps.Checkbox('Novo', key='botaonovo'), ps.Checkbox('Usado', key='botaousado')],
21        [ps.Text('\n')],
22        [ps.Button('Concluir', size=(100,5), font=fontt)],
23        [ps.Text('\n')],
24        [ps.Text('Register(0.0.1)---> Feito por: Anonmixs Amaro Afonso')]
25    ]
26    return ps.Window('Registrador de Livros', layout, finalize=True, size=(400,600))
27
28
29 # Janela opcional(se o livro for usado)
30 def janela_livro_usado():
31     ps.theme('DarkBlue2')
32     layout = [
33         [ps.Text('Escreva um breve resumo do estado de conservação do livro')],
34         [ps.Multiline('Usado, descrição do estado: ', key='livrousoado', size=(400,40))],
35         [ps.Button('Enviar')]
36     ]
37     return ps.Window('Descrição da condição do livro', layout, finalize=True, size=(400,600))
38
39
40 # Janela opcinal do resumo
41 def janela_de_resumo():
42     ps.theme('DarkBlue2')
43     layout = [
44         [ps.Text('##### Resumo #####')],
45         [ps.Output(size=(400,40))],
46         [ps.Button('Fechar o resumo')]
47     ]
48     return ps.Window('Resumo', layout, finalize=True, size=(400,600))
49

```

Figura 2: Definição de cada janela como uma função a ser chamada

Logo após as janelas serem chamadas e devidamente preenchidas, será necessário cadastrar cada entrada(valor) que estão na forma de dicionários em um arquivo .txt para futuras consultas.

Para criar o arquivo .txt mencionado usou-se do comando:

arquivo = open('nome_do_arquivo', 'opção_de_escrita'), que cria ou lê o arquivo se o mesmo já exista, colocando cada entrada digitada no mesmo arquivo .txt.

Capítulo 4

Programa Finalizado

4.1 Alguns Prints do Projeto Finalizado

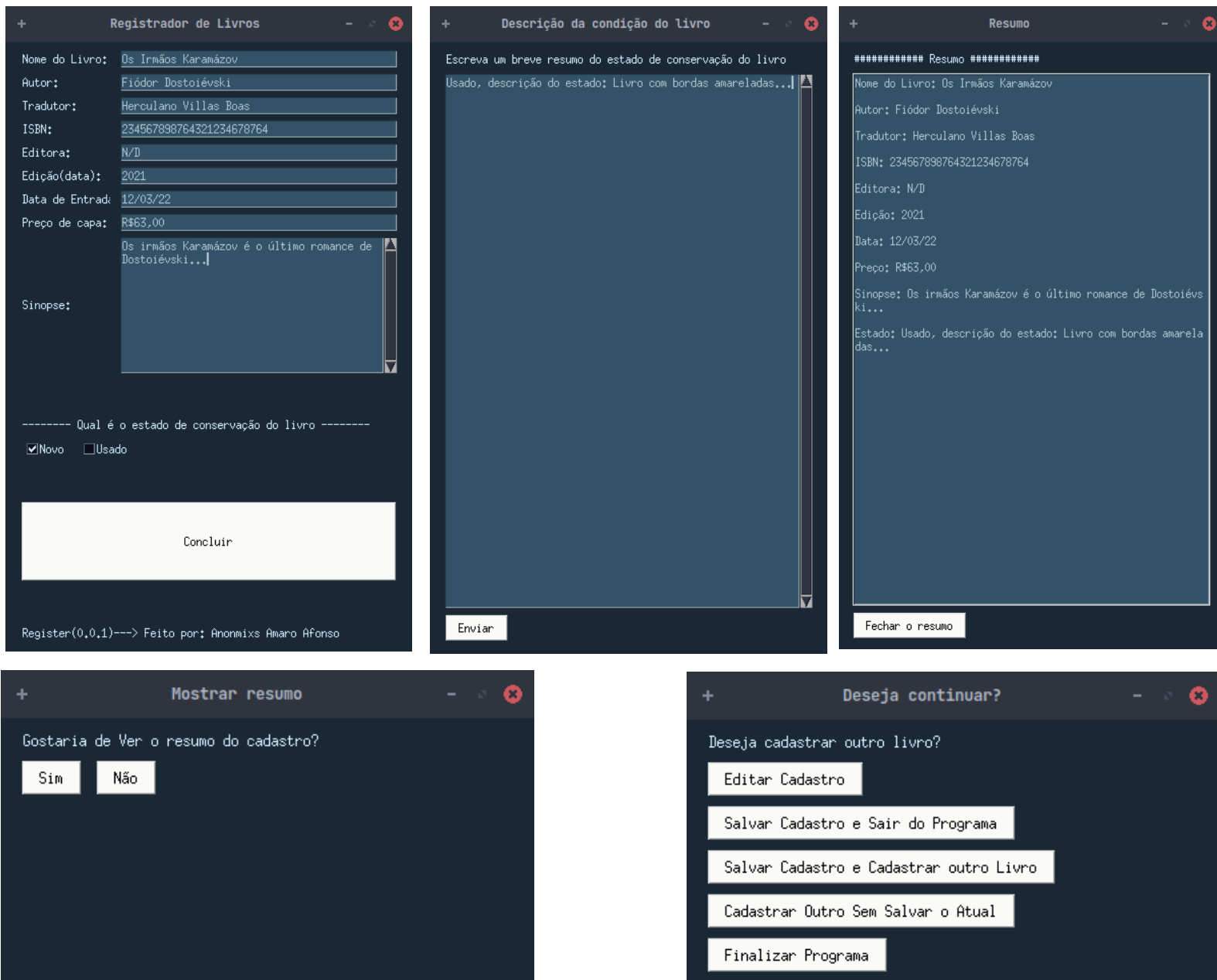
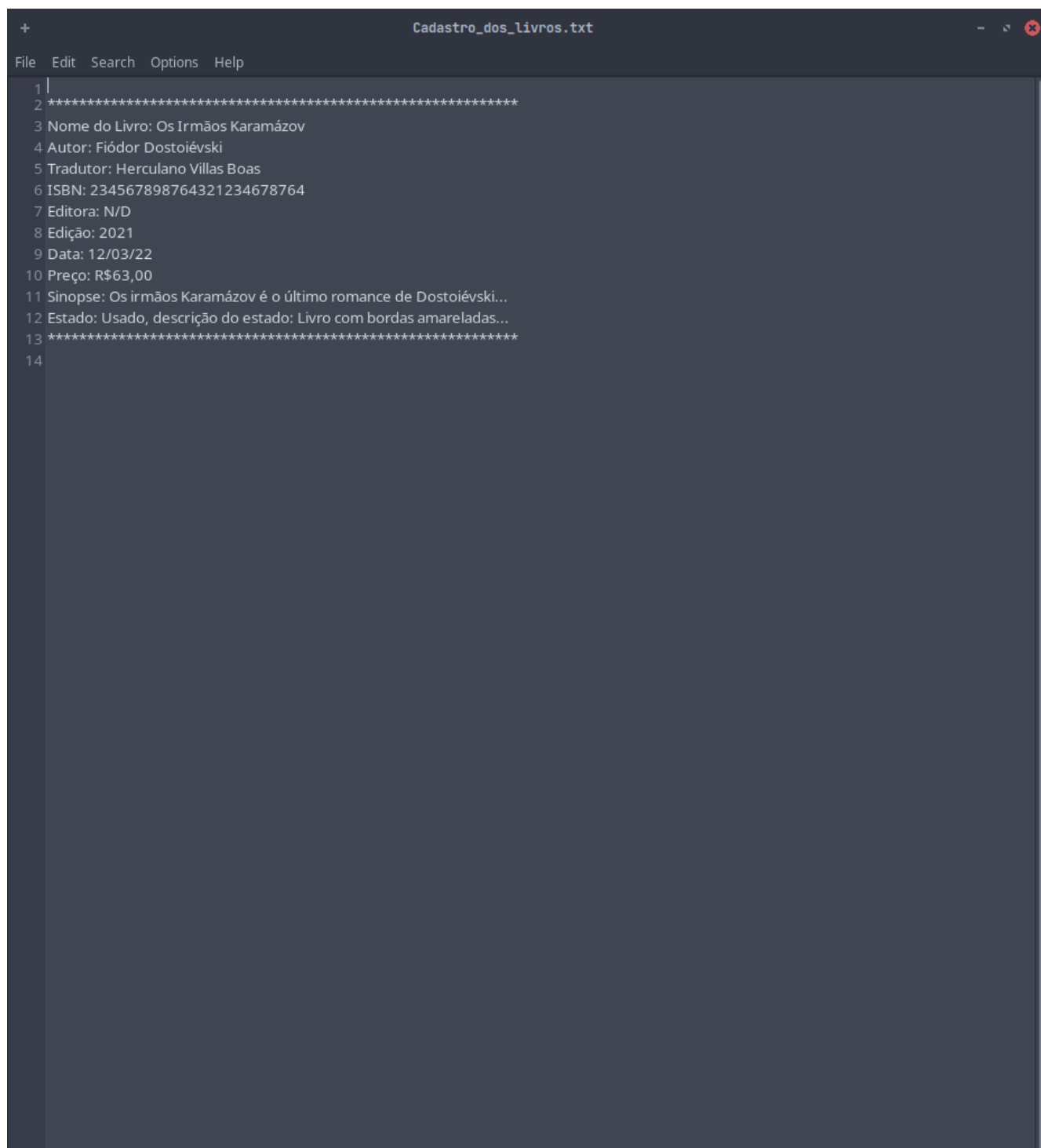


Figura 3: Janelas de cadastro do programa



```
1 |
2 *****
3 Nome do Livro: Os Irmãos Karamázov
4 Autor: Fiódor Dostoiévski
5 Tradutor: Herculano Villas Boas
6 ISBN: 234567898764321234678764
7 Editora: N/D
8 Edição: 2021
9 Data: 12/03/22
10 Preço: R$63,00
11 Sinopse: Os irmãos Karamázov é o último romance de Dostoiévski...
12 Estado: Usado, descrição do estado: Livro com bordas amareladas...
13 *****
14
```

Figura 4: Saída do programa(arquivo .txt) para futuras consultas

Capítulo 5

Conclusão do Projeto

5.1 Implementação do Projeto em uma livraria ou biblioteca de pequena escala

Com o programa finalizado e testado observou-se que o mesmo é muito útil para micro e pequenas empresas no ramo de livros(público alvo desde o começo), onde ficou claro a facilidade do uso e estabilidade do programa, bem como a utilidade da saída do mesmo(no caso o arquivo .txt). Poderíamos por exemplo, utilizando-se de uma linguagem de consulta como a linguagem SQL ou por meio de planilhas com o Excel, construir um banco de dados com todos os livros cadastrados no programa, facilitando as buscas pelos títulos bem como a organização de todo o material disponível na empresa que o utiliza. Tudo isso poderia ser feito de forma totalmente automática, poupando gastos desnecessários com a organização e buscas dos livros bem como deixando tais processos extremamente mais otimizados e precisos.

Vale lembrar que todas as entradas do programa tais como: “Nome do Livro”, “Autor”, “Edição” etc, são customizáveis sendo fácil excluir as atuais ou implementar novas como, qual estante o livro cadastrado se encontra incluindo o número do corredor e número da prateleira. Portanto pode-se criar um programa totalmente customizável para a empresa que obté-lo, mostrando a flexibilidade do código feito.