

Projeto final de nivelamento

Sobre o projeto

A inspiração para o projeto foi um script que eu utilizava em um de meus empregos. Nós da TI utilizávamos um shell chamado "canivete baiano", que tinha diversas funções práticas para executar no servidor. A ideia central foi "Como consumir uma API", mas durante o processo elenquei outras questões para guiar o processo:

- Como "imprimir" imagens na tela?
- Como usar tkinter, suas funcionalidades e fazer GUI?
- Como fazer um programa limpo, sem códigos desnecessários?
- Como criar interfaces bonitas em Python?
- Como utilizar alguma biblioteca para mostrar imagens na tela com python?
- Como utilizar o virtual environment do Python?

Na execução do projeto, acabei mudando o módulo de interface gráfica do Tkinter para o PyQt, pois, além de já ter trabalhado com o mesmo, acho ele mais visual (devido ao QtDesigner), prático de programar e tem como resultado interfaces mais bonitas e amigáveis. Adicionei também a questão do virtual environment, pois era algo que sempre via os programadores fazendo, mas nunca tinha feito. Havia colocado a questão de fazer um instalável do programa, mas percebi que os pacotes para fazer isso não lidam muito bem com projetos baseados em módulos e com caminhos relativos, decidi não prosseguir com a ideia. O projeto foi desafiante, aprendi conceitos novos e consegui desenvolver um código limpo. Gostei muito de trabalhar com o virtual environment, ele facilitou muito a portabilidade do código. Também foi um aprendizado interessante lidar com imagens no PyQt5, o que acabou mostrando a capacidade da biblioteca. Foi implementado o básico de tratamento de erros, que é exibir o traceback para o suporte e uma mensagem de erro para o usuário, sem derrubar o processo ou a interface gráfica, mas também sem diferenciar os tipos de erros gerados pelo código.

Como usar

instalação

Para começar a utilizar o projeto, deve-se instalar os pacotes que são dependência, utilizando o seguinte comando. No Windows:

```
pip install -r .\requirements.txt
```

No Linux:

```
pip install -r requirements.txt
```

Após isso o gerenciador de módulos do Python irá instalar os pacotes necessários. Basta, então, abrir o arquivo "main.py" através do explorador de arquivos ou do terminal. Na pasta do módulo encurtador deve haver um arquivo chamado "variaveis.env", que armazena as variáveis de ambiente necessárias pelo módulo. Dentro desse arquivo deve-se inserir o token obtido no site tinyurl.com. Pode-se utilizar como exemplo o arquivo "variaveis.env.exemplo" e a seguinte API KEY: "P1DwVEQgLd46UFo6XdVl90MV7Ni3WiReRE90p9MgcavSm49Q0B3PMasvb84Y". Caso algo não funcione, entrar em contato!

utilização

O programa é composto por três abas: Ping, Gerar senha e Encurtar URL.

Tente "pingar" ou encurtar o "alegrete" =>

PING

Na aba de ping é onde se executa o comando de ping, para verificar se um host está online (obs: o host pode não responder ao ping e estar online). Para isso insira o host a ser "pingado", a quantidade de vezes que o ping deve ser executado e clique no botão para executar o ping. A resposta será mostrada em dois locais, na etiqueta abaixo do host, que dirá se o host respondeu ou não ao comando e no descritor avançado do ping, que mostra a saída do processo.

GERAR SENHA

Nessa aba é gerada uma senha randômica e forte, com tamanho de 8 a 64 caracteres. Para isso deve-se deslizar os sliders indicadores de tamanho de senha e de quantidade de caracteres especiais, ao clicar no botão de geração, ela irá aparecer na caixa de texto e será copiada para a área de transferência.

ENCURTAR LINK

Aqui é feito o encurtamento de uma url longa, por exemplo:

"<https://www.example.com/my-really-long-link-that-I-need-to-shorten/84378949>" em algo como: "<https://tinyurl.com/29btt3a5>". Ao preencher o caminho a ser encurtado e clicar em gerar, é gerado um link e um QRCode que aponta para esse link encurtado. O QRCode também é salvo na pasta qrcodes, com nome qr_XXX_YYY.