Envio de Promoções no Wpp

Este código em Python utiliza a biblioteca Tkinter para criar uma interface gráfica e realizar operações em um banco de dados SQLite. Ele define a classe Funções, que contém métodos para conectar e desconectar-se ao banco de dados, criar tabelas, selecionar e buscar informações e limpar a tela. A classe Up\_Down herda da classe Funções e adiciona os métodos open, verifica e Upload para permitir a abertura de um arquivo, verificação de nomes de clientes no banco de dados e upload de dados para o banco de dados.

A classe "funcoesClientes" é uma subclasse da classe "Funcoes" e contém métodos para manipulação de dados relacionados aos clientes em um sistema de gerenciamento de negócios. Os métodos incluem a adição, exclusão e edição de clientes, bem como o envio de mensagens para eles.

**Métodos:**

* variaveis\_clientes(): atribui os valores inseridos nos campos de entrada a variáveis correspondentes.
* add\_cliente(): insere um novo cliente na tabela "clientes" do banco de dados, caso o nome e o telefone não estejam vazios.
* abre\_cliente(): exibe as informações de um cliente selecionado na tela para edição.
* deleta\_cliente(): exclui um cliente da tabela "clientes" do banco de dados, com base em seu ID.
* editar\_cliente(): atualiza as informações de um cliente na tabela "clientes" do banco de dados, com base em seu ID.
* envia\_msg(): envia uma mensagem para todos os clientes na tabela "clientes" do banco de dados, usando a classe "autoMsg".

Os seguintes atributos são usados nos métodos:

* entrada\_id: campo de entrada para o ID do cliente.
* entrada\_nome: campo de entrada para o nome do cliente.
* entrada\_email: campo de entrada para o endereço de e-mail do cliente.
* entrada\_data1: campo de entrada para a data da primeira compra do cliente.
* entrada\_cpf: campo de entrada para o CPF do cliente.
* entrada\_wpp: campo de entrada para o número de telefone do cliente.
* entrada\_mensagem: campo de entrada para a mensagem a ser enviada aos clientes.
* saida: exibe as informações dos clientes na tela.
* conexao: objeto de conexão com o banco de dados.
* cursor: objeto de cursor para executar comandos SQL no banco de dados.
* msg: a mensagem a ser enviada aos clientes.
* autoMsg: uma classe externa para enviar mensagens aos clientes.

A classe Application define uma aplicação GUI para enviar promoções a clientes cadastrados. Ela herda os atributos e métodos das classes funcoesClientes e Up\_Down. Ela possui os seguintes métodos:

**\_\_init\_\_(self)**

* Construtor da classe que inicializa a aplicação, definindo os métodos de personalização da tela, criação de widgets e seleção de saída. Por fim, inicia o loop principal da aplicação.

**personalisacao(self)**

* Método que personaliza a aparência da tela da aplicação. Os parâmetros são as cores de fundo da tela, de fundo do botão, da borda do botão e da cor do texto do botão, o tamanho da fonte, o tipo da fonte (negrito, itálico, normal), entre outros.

**tela(self)**

* Método que define as características básicas da janela da aplicação, tais como título, tamanho e cor de fundo.

**frames\_tela\_cadastro(self)**

* Método que cria dois frames na tela: um para o cadastro de clientes e outro para a exibição de clientes cadastrados. Esses frames são configurados com cor de fundo e borda.

**paginas(self)**

* Método que cria as abas da aplicação, uma para cadastro e outra para envio de promoções. As abas são adicionadas ao widget ttk.Notebook.

**botoes(self, aba)**

* Método que cria os botões de limpar, buscar, adicionar, editar e apagar. Esses botões são adicionados à aba correspondente.

**widgets\_pag\_1(self)**

* Método que cria os widgets da aba de cadastro, como as entradas de dados e rótulos. Além disso, ele chama o método botoes para criar os botões.

**widgets\_pag\_2(self)**

* Esse método é responsável por criar os widgets da segunda aba da aplicação. A segunda aba é a aba de envio de mensagem.

**output\_frame\_2(self)**

* Esse método é responsável por criar os widgets da segunda frame da aplicação. A segunda frame é a frame que contém a tabela de clientes.

Menus(self)

* Esse método é responsável por criar os menus da aplicação.
* O método cria dois menus principais, "Opções" e "Ajuda", e os adiciona na barra de menu da aplicação.
  + O menu "Opções" possui três opções:
    - A opção "Upload" chama o método Upload da classe Application, que permite ao usuário fazer upload de um arquivo.
    - A opção "Download" permite ao usuário fazer download de um arquivo.
    - A opção "Sair" chama o método destroy da classe root, que fecha a janela da aplicação.
  + O menu "Ajuda" possui apenas uma opção, "Modo de Usar", que chama a função Quit.

AutoBot