DESENVOLVIMENTO DE UMA APLICAÇÃO EM PYTHON PARA AUTOMATIZAR PROCESSOS DE PROSPECÇÃO ATIVA

Cecir Barbosa de Almeida Farias (UFCG-CDSA) cecir.almeida@gmail.com.br

Mateus José De Siqueira Silva (UFCG-CDSA) mateussiqueirasilva3@gmail.com

Pedro Paulo Mendes Tomaz (UFCG-CDSA) pedrtomz@gmail.com

Resumo

A pesquisa tem como objetivo demonstrar as sugestões de melhorias diagnosticadas após a aplicação das ferramentas da administração Análise SWOT, Matriz BCG e as Cinco Forças de Porter, em uma empresa do ramo de manutenção automobilística situada na Cidade de Campina Grande, no estado da Paraíba, com intuito de identificar a atual situação da empresa no mercado no qual está inserida. Para construção desse trabalho foi realizada uma entrevista semiestruturada com o gestor geral da referida empresa (objeto de estudo). Concluiu-se que a empresa possui serviços de excelência e com um grande potencial de crescimento, porém, precisa das melhorias referidas no artigo.

Palavras-Chaves: Estratégia. Ferramentas Organizacionais. Microempresas. Análise SWOT. Matriz BCG. Cinco Forças de Porter. Manutenção Automotiva.

1. Introdução

Para organizações que verdadeiramente buscam perspectivas de sucesso, é imprescindível o uso de ferramentas administrativas descritas no presente trabalho, que visa maximizar o potencial escalável da empresa objeto de estudo através de tais artifícios. Segundo Almeida citado por Faller e Almeida (2014), os fatores que mais geram dificuldades na manutenção da competitividade e a sobrevivência não são as ocasionadas por insuficiência de recursos dessas empresas e sim os relacionados a fatores estratégicos.

Em um cenário global que evidencia uma realidade empresarial dinâmica, as inovações são indispensáveis para essas organizações, levando em consideração o fato de que dependem de resultados positivos para que ocorra a sua manutenção no mercado, e também o seu crescimento constante, ancorado pelo almejado diferencial competitivo (Aff & de Araújo, 2013). Para isso foram aplicadas ferramentas administrativas numa empresa no ramo da

manutenção automotiva com o objetivo de fundamentar as decisões dos tomadores de decisão de dar ênfase nos pontos críticos da organização.

A pesquisa é qualitava de caráter exploratório e encontra-se estruturada em cinco capítulos onde, no segundo, apresenta-se o referencial teórico, iniciando de estratégias da administração e estratégias aplicadas em pequenas empresas, passando por ferramentas abordadas neste artigo; no terceiro se destacam os procedimentos metodológicos utilizados; no quarto são descritos os resultados e discussões e no capítulo final apresenta-se a conclusão da pesquisa dentro dos alinhamentos gerais e específicos da mesma.

2. Referencial Teórico

2.1. Banco de Dados

Banco de dados são coleções ou conjuntos de arquivos e informações organizadas e estruturadas, que são armazenados em um sistema de um computador com o objetivo de manutenção de registros (Date, 2004). Os Bancos de Dados evoluíram exponencialmente desde da sua criação no início dos anos 1960 desenvolvido na empresa IBM, pelo pesquisador Edgar Frank, o que possibilitou os usuários armazenarem e extraírem grandes quantidades de informações.

O principal objetivo dos bancos de dados é facilitar o fornecimento de informações aos usuários de um sistema. As principais vantagens dos Bancos de Dados referem-se ao alto nível de estrutura e suportes consolidados da computação, além do mais, são fáceis de armazenar e de serem recuperados. Os principais Bancos de Dados são: *Oracle, SQL Server, MySQL, PostgreSQL, DB2, NoSQL, MongoDB, Redis, InfluxDB* e *DynamoDB.*

2.2. *SQLite*

O *MySQL* é capaz de armazenar qualquer tipo de informação, desde um *site* institucional até um *E-commerce*, incluindo desde o registro de um simples dado até todo o inventário de produtos. Seu modelo de criação de Banco de Dados é baseado em tabelas. Um Banco de Dados pode conter muitas tabelas para organizar as informações de uma forma mais concisa. (PISA, 2012).

SQLite é uma biblioteca de código aberto (*open source*) desenvolvido na linguagem C que permite a disponibilização de um pequeno Banco de Dados na própria aplicação, sem a necessidade de acesso a um SGDB - Sistema Gerenciador de Banco de Dados separado. A

grande vantagem desse tipo de Banco de Dados está em sua simplicidade e praticidade de implementar e administrar seus dados, utilizando soluções como o *SQL Server* e o *Oracle*.

O uso do *SQLite* é recomendado para:

- Aplicativos básicos *desktop* / *mobile*
- Pequenos *Websites*
- Sistemas de informação utilizados por poucas pessoas

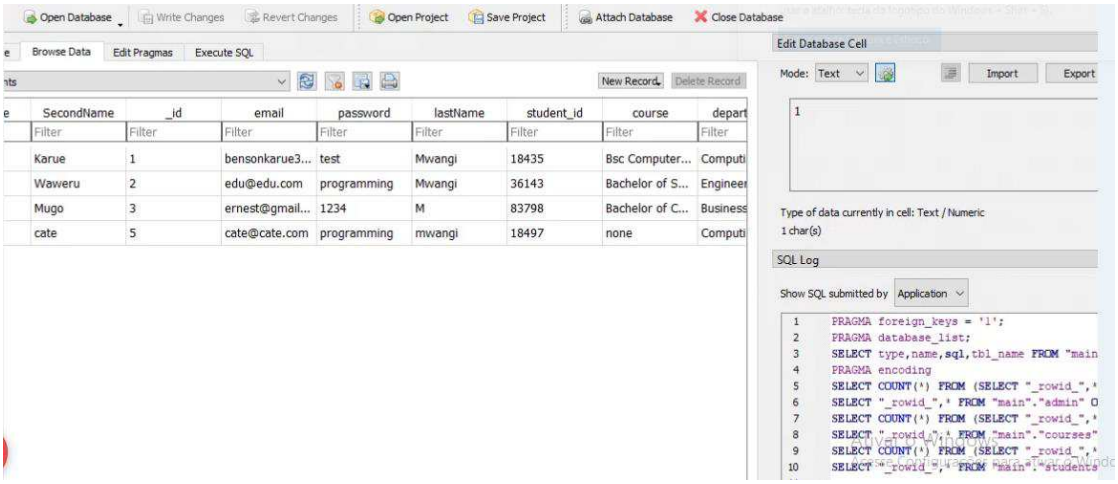
2.3. *Python*

A Linguagem *Python* foi concebida no fim dos anos 80. A primeira ideia de implementar o *Python* surgiu mais especificamente em 1982 enquanto *Guido Van Rossum* trabalhava no CWI - *Centrum Wiskunde & Informatica* (Centro de Matemática e Ciência da Computação) em Amsterdã, Holanda, no time de desenvolvimento da Linguagem ABC. Posteriormente, em 1987, com o fim da linguagem ABC, Guido foi transferido para o grupo de trabalho *Amoeba* com um sistema operacional *Microkernel* liderado por *Andrew Tanenbaum*. Foi neste grupo que Guido percebeu a necessidade de uma linguagem para escrever programas intermediários, algo entre o C e o *Shell Script*. Em 1989 o desenvolvimento do *Python* realmente teve início. Nos primeiros meses de 1990 o autor já possuía uma versão mínima e operacional, pelo fim do ano de 1990 *Python* já era mais utilizada no CWI que a própria linguagem ABC (Mind Bending, 2014).

2.4. DB Browser

O *DB Browser for SQLite* permite visualizar e editar Bancos de Dados *SQLite* no *Windows*. Pode-se projetar, criar e editar esses arquivos de Banco de Dados (UNIVASF, 2020). Através do *DB Browser* é possível criar, visualizar e editar Bancos de Dados, de forma que não seja necessário usar linhas de códigos em um ambiente de programação para o mesmo. Tais funcionalidades podem ser executadas na aba "Executar SQL". A ilustração do *DB Browser* poder ser vista na Figura 1.

Figura 1 – *DB Browser*



Fonte: Disponível em: HACKSMILE (2019).

3. Metodologia

Nesta seção apresentam-se os métodos (preparação e planejamento) referentes à realização da pesquisa, para de forma clara e objetiva, poder elucidar a maneira como foi constituído o presente trabalho.

O objeto de estudo selecionado foi uma empresa júnior do ramo de consultorias e soluções organizacionais, cujo nome fantasia é PRODUP, localizada na cidade de Sumé no estado da Paraíba, com vínculo ao campus universitário do CDSA (Centro de desenvolvimento Sustentável do Semiárido) da UFCG (Universidade Federal de Campina Grande). A escolha se deu devido a necessidade de otimização do envio de *e-mails* para prospecção ativa da empresa.

As ferramentas utilizadas para desenvolvimento do código que possibilita a ação automatizada dos *e-mails* foram: *DB Browser* para SQL - ferramenta responsável pela criação do Banco de Dados de *leds*, com suas respectivas informações desejadas pela empresa, o *IDLE* - ambiente de desenvolvimento para *Python* e essa linguagem de programação para criação dos códigos.

O presente trabalho apresenta uma pesquisa aplicada, ou seja, aquela cujo principal objetivo é a geração de conhecimento para aplicação prática e imediata, dirigidos à solução de problemas específicos envolvendo os interesses locais, territoriais e regionais (IFPA 2019).

4. Resultados e Discursões

4.1. Descrição da Empresa

A empresa objeto de estudo é do ramo de Engenharia de Produção, composta por estudantes de Engenharia de Produção do CDSA, uma empresa de porte médio no cenário de empresas juniores do Brasil. Seu ramo de atuação é voltado para consultorias e soluções organizacionais, com atuação em todos os ramos de negócios de Sumé e região.

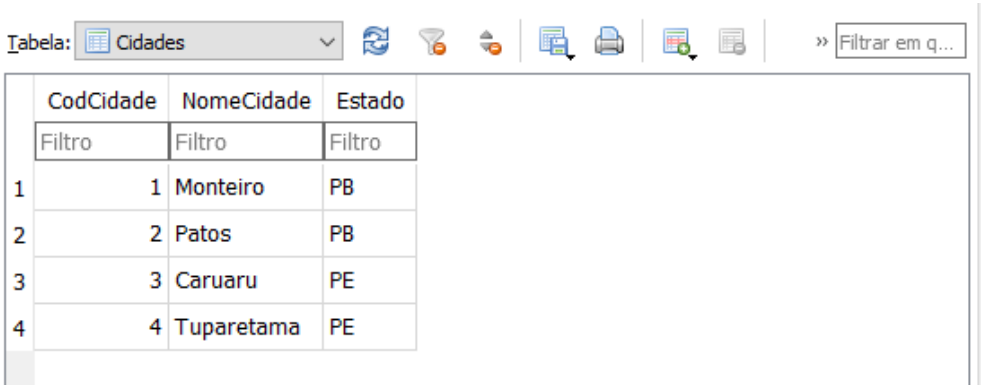
4.2. Banco de Dados SQLite

Os elementos utilizados para a criação do Banco de Dados no *DB Browser for SQL*, foram provenientes do trabalho de mapeamento de *leds* realizado pela empresa júnior. Neste Banco de Dados foram adicionadas duas tabelas cujos nomes são "Cidades" e "Informação".

Na tabela "Cidades", foram adicionadas 4 cidades mapeadas pela empresa júnior: Monteiro, Patos, Caruaru e Tuparetama.

Os dados estão representados na Figura 2.

Figura 2 – Tabela Cidades



	CodCidade	NomeCidade	Estado
	Filtro	Filtro	Filtro
1	1	Monteiro	PB
2	2	Patos	PB
3	3	Caruaru	PE
4	4	Tuparetama	PE

Fonte: Autoria Própria (2021)

Na tabela "Informação" foram adicionados os dados necessárias para um primeiro contato através da prospecção ativa, de cada um dos *leds* presentes nessas cidades de interesse da PRODUP. Seus dados podem ser visualizados na Figura 3.

Figura 3 – Tabela

Informação

Tabela: Informacao — Filtrar em qualquer coluna

	CodLoja	CodCidade	NomeEmpresa	Seguimento	DordaEmpresa	RazãoSocial	Email
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	11	1	Empório Laffê	Vestuario	Estoque	Maria Alany	lannyoliveira.01@hotmail.com
2	12	1	Combate Sports	Vestuario	Estoque	Viviane Patriota	lojacombatsport@gmail.com
3	13	1	Max Modas Liberty	Vestuario	Estoque	Maria Laudilene	maxmodas1@hotmail.com
4	14	1	Lojas Olindina	Vestuario	Marketing	Antonio Aprigio	antonioaprigio@yahoo.com.br
5	15	1	PONTO CHIK SUME	Vestuario	Estoque	Maria Renilda	renilasoares@gmail.com
6	16	1	Mamãe eu quero	Vestuario	Marketing	Leandro Lucas	vanessac_tome@yahoo.com.br
7	21	2	PUKET	Vestuario	Marketing		puket.guedesshopping@gmail.com
8	22	2	Li Mont	Vestuario	Marketing		li.mont@outlook.com
9	23	2	Loja Stillo	Vestuario	Marketing	Maria	elianestillo@hotmail.com
10	24	2	Loja AZ	Vestuario	Marketing		arize_almeida@hotmail.com
11	31	3	Pijamas Caruaru	Vestuario	Marketing	Erica	erica.cagoes@gmail.com
12	32	3	CARUARU MULTMARCAS	Vestuario	Marketing	Jhenifa	jhennifaalbino17@hotmail.com
13	33	3	Litoraneos Caruaru	Vestuario	Marketing	Julio	juliopinel@gmail.com
14	34	3	Vitória Moda Íntima	Vestuario	Marketing	Vitoria	vitoriamoda001@gmail.com
15	35	3	Molecagem Moda Infantil	Vestuario	Marketing	Paulo	paulo.fco@hotmail.com
16	41	4	LOJAS SURYA SÃO JOSÉ	Vestuario	Estoque	Surya	suryasaojose1@hotmail.com
17	42	4	Outlet Destaque	Vestuario	Marketing	José Rodrigues	destackkk@hotmail.com
18	43	4	Moderna Calçados	Vestuario	Marketing	Ana Cristina	betoleitecont@yahoo.com.br

1 - 19 de 19 — Ir para: 1

Fonte: Autoria Própria (2021)

4.3. Desenvolvimento Dos Códigos

Para desenvolvimento dos códigos na linguagem P*ython*, foi utilizado o ambiente de programação *IDLE*. Levando-se em conta a necessidade da empresa júnior de automatizar o processo de prospecção ativa por meio do envio de e-mails, foi desenvolvida esta aplicação.

Para seu desenvolvimento foi necessário importar cinco bibliotecas do *Python* como mostra a Figura 4.

Figura 4 – Bibliotecas importadas

projeto_teste.py - C:\Users\CCE info\Downloads\projeto_teste.py (3.9.6)
File Edit Format Run Options Window Help

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import sqlite3 as sq
from datetime import datetime
from os import path
```

Fonte: Autoria Própria (2021)

Para conectar com o servidor do serviço de e-mail e construir o corpo, foram importadas as bibliotecas "*smtplib", "MIMEMultipart"* e *"MIMEText*". O disparo automático dos e-mails é

realizado com auxílio da biblioteca "*datetime*" que tem por função carregar o horário atual do computador para que, quando o mesmo equalizar com os horários definidos de envio, a operação seja executada. A biblioteca "path" é responsável por conectar o código ao banco de dados juntamente com a biblioteca "sqlite3" fazendo com que operações possam ser realizadas no banco.

Posteriormente a importação das bibliotecas, iniciou-se o processo de automatização do envio de e-mails em datas pré-definidas pelo usuário da aplicação, estas datas foram definidas pelo diretor comercial da empresa.

Para entrar no servidor da *Google* é necessário ter ciência das informações de *host* e *port* que podem ser encontradas nas configurações de conta Google. Código descrito na Figura 5.

Figura 5 – Conexão com servidor Google

Fonte: Autoria Própria (2021)

Para fazer o carregamento dos contatos a serem enviados para cada cidade, o algoritmo seleciona os contatos e enxerta em uma lista provisória (lista_*email*) com mostra a Figura 6.

Figura 6 – Processo de carregamento de contatos do banco de dados

```
#email para Monteiro
def carregar_contatos_monteiro():
    cursor.execute("""SELECT Email FROM Informacao WHERE CodCidade = 1""")
    for linha in cursor.fetchall()  :
        for palavra in linha :
            lista_email.append(palavra)

#email para Patos
def carregar_contatos_patos():
    cursor.execute("""SELECT Email FROM Informacao WHERE CodCidade = 2""")
    for linha in cursor.fetchall() :
        for palavra in linha :
            lista_email.append(palavra)

#email para Caruaru
def carregar_contatos_caruaru():
    cursor.execute("""SELECT Email FROM Informacao WHERE CodCidade = 3""")
    for linha in cursor.fetchall() :
        for palavra in linha :
            lista_email.append(palavra)

#email para Tuparetema
def carregar_contatos_tuparetama():
    cursor.execute("""SELECT Email FROM Informacao WHERE CodCidade = 4""")
    for linha in cursor.fetchall() :
        for palavra in linha :
            lista_email.append(palavra)

#email para Sumé
def carregar_contatos_sume():
    cursor.execute("""SELECT Email FROM Informacao WHERE CodCidade = 5""")
    for linha in cursor.fetchall() :
        for palavra in linha :
            lista_email.append(palavra)
```

Fonte: Autoria Própria (2021)

O carregamento das datas de envio de cada cidade é feito por uma busca no Banco de Dados e armazenado em uma variável provisória como descrito na Figura 7.

Figura 7 – Carregamento das datas de envio

```
#datas de envio
#Monteiro
cursor.execute("""SELECT DataEnvio FROM Cidades WHERE CodCidade = 1""")
for linha in cursor.fetchall() :
    for data in linha :
        data_de_envio_Cid_1 = data

#Patos
cursor.execute("""SELECT DataEnvio FROM Cidades WHERE CodCidade = 2""")
for linha in cursor.fetchall() :
    for data in linha :
        data_de_envio_Cid_2 = data

#Caruaru
cursor.execute("""SELECT DataEnvio FROM Cidades WHERE CodCidade = 3""")
for linha in cursor.fetchall() :
    for data in linha :
        data_de_envio_Cid_3 = data

#Tuparetama
cursor.execute("""SELECT DataEnvio FROM Cidades WHERE CodCidade = 4""")
for linha in cursor.fetchall() :
    for data in linha :
        data_de_envio_Cid_4 = data

#Sumé
cursor.execute("""SELECT DataEnvio FROM Cidades WHERE CodCidade = 5""")
for linha in cursor.fetchall() :
    for data in linha :
        data_de_envio_Cid_5 = data
```

Fonte: Autoria Própria (2021)

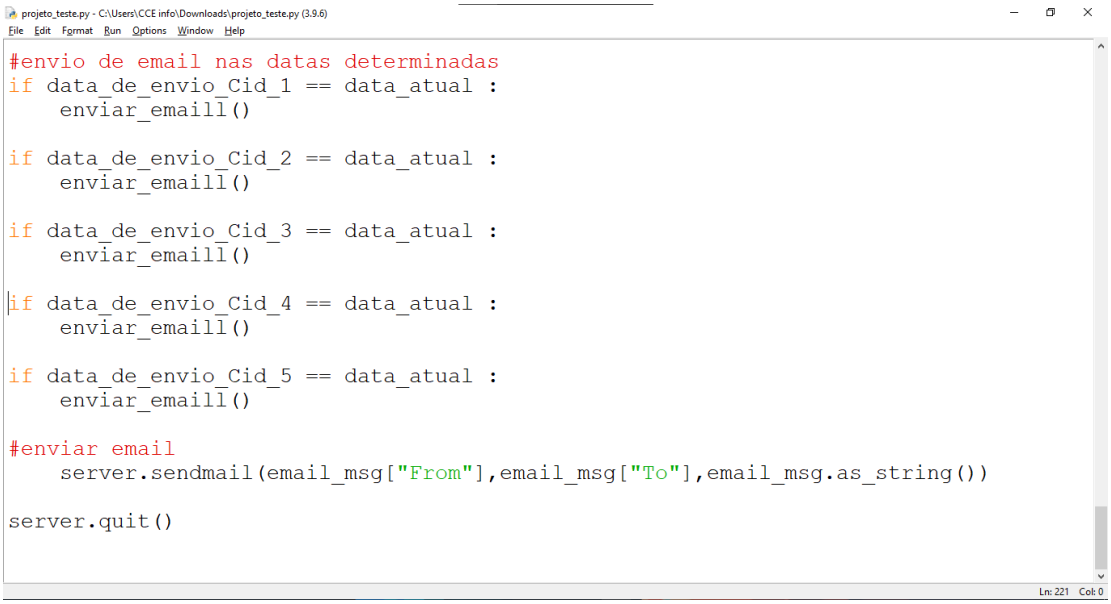
Para que seja escrito o *e-mail* no formato necessário é utilizada a biblioteca *MIMEMultipart* para que seja reconhecido pelo *Gmail* conforme é mostrado na Figura 8.

Figura 8 - Processo de automação do envio de e-mails

Fonte: Autoria Própria (2021)

O e-mail é enviado através de um algoritmo que sincroniza a data e hora atual do computador em que se está sendo rodado o programa com a data presente no Banco de Dados, sendo então chamada a função "enviar_*email*" que executa os disparos de envios.

Figura 9 – Sincronização das datas determinadas com a data atual.



Fonte: Autoria Própria (2021)

Em seguida, criou-se o *menu* interativo com o usuário, contendo as funções de:

[0] Sair;

[1] Definir a data para enviar *e-mail*;

[2] Ver lista de contatos;

[3] Deletar contato;

[4] Exibir lojas em na ordem das cidades;

[5] Atualizar contato;

[6] Ver contatos da cidade de Monteiro e Patos;

[7] Ver contatos por cidade;

[8] Exibir datas de envio.

Este *menu* possibilita ao usuário diferentes funcionalidades com a execução do programa e com os dados presentes em seu banco de informações. Foi utilizado o comando "*While*" para que o *menu* seja sempre reapresentado após cada operação realizada e, dentro das opções contém outras opções de interatividade com o usuário. Se o número selecionado estiver entre as opções os comandos "if" serão executados conforme descrito na Figura 11, Figura 12, Figura 13, Figura 14 e Figura 15.

Figura 11 - Código *menu* interativo

```
#menu
while (True):
    #definir data
    print(" [0] Sair","\n","[1] Definir a data para enviar email","\n","[2] Ver lista de Contatos","\n","[3] Deletar conta
    opcao = int(input(""))

    if opcao == 1 :

        #selecionar cidade
        while (True):
            print(" [1] Monteiro","\n", "[2] Patos","\n","[3] Caruaru","\n","[4] Tuparetama","\n","[5] Sumé","\n")
            opcao_cidade = int(input())

            if opcao_cidade == 1 :
                print("Digite a data que deseja enviar o email (Favor digitar com / )")
                set_data = str(input())
                cursor.execute("""UPDATE Cidades SET DataEnvio = (?) WHERE CodCidade = 1""",[set_data])
                break

            elif opcao_cidade == 2 :
                print("Digite a data que deseja enviar o email (Favor digitar com / )")
                set_data = str(input())
                cursor.execute("""UPDATE Cidades SET DataEnvio = (?) WHERE CodCidade = 2""",[set_data])
                break

            elif opcao_cidade == 3 :
                print("Digite a data que deseja enviar o email (Favor digitar com / )")
                set_data = str(input())
                cursor.execute("""UPDATE Cidades SET DataEnvio = (?) WHERE CodCidade = 3""",[set_data])
                break
```

Fonte: Autoria Própria

Figura 12 – Continuação do código *menu* interativo

```
            elif opcao_cidade == 4 :
                print("Digite a data que deseja enviar o email (Favor digitar com / )")
                set_data = str(input())
                cursor.execute("""UPDATE Cidades SET DataEnvio = (?) WHERE CodCidade = 4""",[set_data])
                break

            elif opcao_cidade == 5 :
                print("Digite a data que deseja enviar o email (Favor digitar com / )")
                set_data = str(input())
                cursor.execute("""UPDATE Cidades SET DataEnvio = (?) WHERE CodCidade = 5""",[set_data])
                break


    if opcao == 2 :
        cursor.execute("""SELECT * FROM Informacao""")
        for linha in cursor.fetchall() :
            print(linha)

    if opcao == 3 :
        codLoja = int(input("Digite o código da loja para excluir o contato: "))
        cursor.execute("""DELETE FROM Informacao WHERE CodLoja = (?) """,[codLoja])
        print("Operação realizada com sucesso")

    if opcao == 4 :
        #adicionar codCidade ao lado para sinalizar
        cursor.execute("""SELECT NomeEmpresa FROM Informacao ORDER By CodCidade""")
        for linha in cursor.fetchall() :
            print(linha)
```

Fonte: Autoria Própria

Figura 13 - Continuação do código *menu* interativo

```
        elif opcao_cidade == 5 :
            print("Digite a data que deseja enviar o email (Favor digitar com / )")
            set_data = str(input())
            cursor.execute("""UPDATE Cidades SET DataEnvio = (?) WHERE CodCidade = 5""",[set_data])
            break


if opcao == 2 :
    cursor.execute("""SELECT * FROM Informacao""")
    for linha in cursor.fetchall() :
        print(linha)

if opcao == 3 :
    codLoja = int(input("Digite o código da loja para excluir o contato: "))
    cursor.execute("""DELETE FROM Informacao WHERE CodLoja = (?) """,[codLoja])
    print("Operação realizada com sucesso")

if opcao == 4 :
    #adicionar codCidade ao lado para sinalizar
    cursor.execute("""SELECT NomeEmpresa FROM Informacao ORDER By CodCidade""")
    for linha in cursor.fetchall() :
        print(linha)

if opcao == 5 :
    print("Digite o código da loja: ")
    codLoja = int(input())
    print("Digite o nome: ")
    update_nome = str(input())
    cursor.execute("""UPDATE Informacao SET RazãoSocial = (?) WHERE CodLoja = (?)""",[update_nome,codLoja])
    print("Operação realizada com sucesso")
```

Fonte: Autoria Própria

Figura 14 – Continuação do código *menu* interativo

```
    if opcao == 6 :
        cidade_opcao_1 = int(input("Digite o código da primeira cidade: "))
        cidade_opcao_2 = int(input("Digite o código da segunda cidade: "))
        cursor.execute("""SELECT * FROM Informacao WHERE CodCidade BETWEEN (?) AND (?);""",(cidade_opcao_1,cidade_opcao_2)
        for linha in cursor.fetchall() :
            print(linha)

    if opcao == 7 :
        while (True) :
            print(" Escolha a cidade","\n","[1] Monteiro","\n", "[2] Patos","\n","[3] Caruaru","\n","[4] Tuparetama","\n",
            choose_cidade = int(input(" "))
            if choose_cidade == 1 :
                carregar_contatos_monteiro()
                print(lista_email)
                break

            if choose_cidade == 2 :
                carregar_contatos_patos()
                print(lista_email)
                break

            if choose_cidade == 3 :
                carregar_contatos_caruaru()
                print(lista_email)
                break

            if choose_cidade == 4 :
                carregar_contatos_tuparetama()
                print(lista_email)
                break
```

Fonte: Autoria Própria

Figura 15 – Continuação do código *menu* interativo

```
if opcao == 8 :
    cursor.execute("""SELECT * FROM Cidades""")
    for linha in cursor.fetchall() :
        print(linha)

if opcao == 0 :
    banco.close
    break

else:
    continue
```

Fonte: Autoria Própria

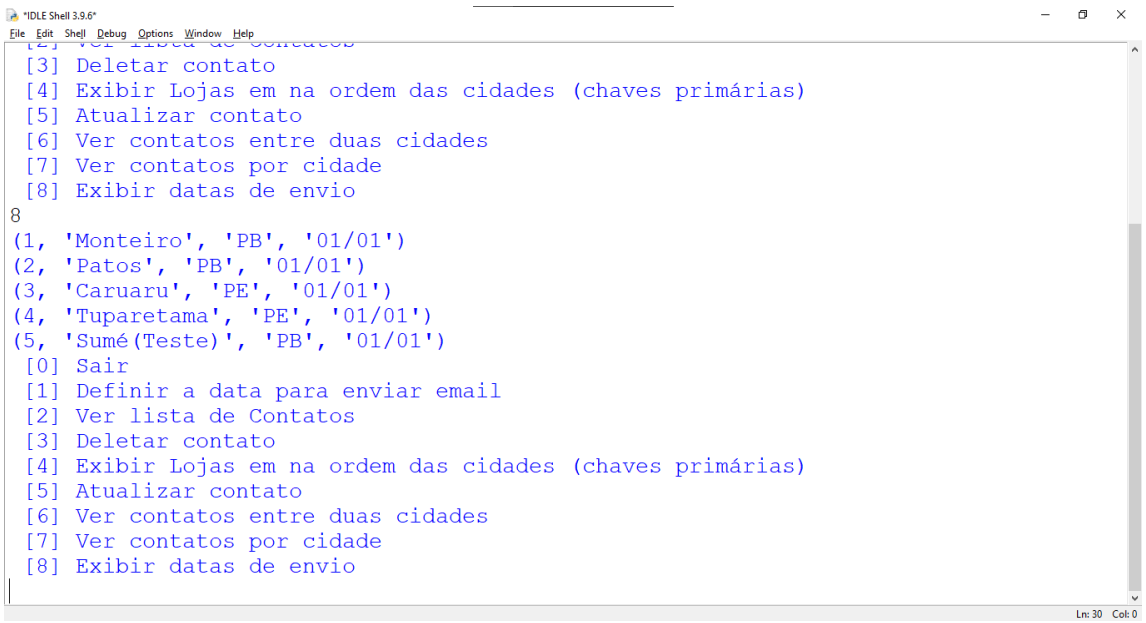
O *menu* interativo mostra nove opções onde o usuário escolherá qual deseja executar de acordo com sua necessidade. Este *menu* pode ser observado na codificação exemplificada na Figura 16 e na Figura 17.

Figura 16 - *Menu* Principal

```
*IDLE Shell 3.9.6*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64
)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\CCE info\Documents\PEDRO\CDSA\TEEP_avancos_em_prog\automacao_em
ail\automacao_email_BETA.py
Por questão de ética, fazer testes apenas para sumé que está configurada para o mes
mo!!
 [0] Sair
 [1] Definir a data para enviar email
 [2] Ver lista de Contatos
 [3] Deletar contato
 [4] Exibir Lojas em na ordem das cidades (chaves primárias)
 [5] Atualizar contato
 [6] Ver contatos entre duas cidades
 [7] Ver contatos por cidade
 [8] Exibir datas de envio
```

Fonte – Autoria próprio

Figura 17 – *Menu* Principal



Fonte – Autoria pória

5. Considerações Finais

O objetivo inicial de automatizar a prospecção ativa da Empresa Júnior de Engenharia de Produção - PRODUP foi alcançado. As funções de: envio autônomo de *e-mails* pré-definidos; inserção de dados entre faixas específicas; obtenção de dados das tabelas com informações dos *leads*; remoção de registros do Banco de Dados, apresentação da tabela de prospectados e suas informações possibilitam um maior controle por parte do gestor comercial quanto a execução do programa e sua estruturação no Banco de Dados.

O código será disponibilizado para a empresa, com possibilidade de conversão em um aplicativo posteriormente, tornando-o usual e possibilitando a aplicação do mesmo em instituições de diferentes ramos que necessitam do mesmo segmento de prospecção para garantia de seus processos internos.

REFERÊNCIAS

AFF, C. C., & DE ARAÚJO, R. M. (2013). Móveis planejados: Um estudo sobre a cadeia de fornecimento no contexto da inovação. Revista Eletrônica do Mestrado Profissional em Administração da Universidade Potiguar.

C. J. Date. Introdução a Sistemas de Bancos de Dados. GEN LTC, 8ª edição. 2004.

O que é SQLite - Portal GSTI. Disponível em: <https://www.portalgsti.com.br/sqlite/sobre/#:~:text=SQLite%20%C3%A9%20uma%20biblioteca%20de,acesso%20a%20um%20SGDB%20separado>. Acesso em: 04 out 2021.

HACKSMILE. How to Manage SQLite Database on PC without Browser Extensions. 2019. Disponível em: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fhacksmile.com%2Fhow-to-manage-sqlite-database-on-windows-without-browser-extensions%2F&psig=AOvVaw3T0UBY0HcExehkCebvi0b_&ust=1633456266626000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCLCL8oOpsfMCFQAAAAAdAAAAABAP.> Acesso em: 03 out 2021.

IFPA. ANEXO II – ROTEIRO DO PRÉ-PROJETO DE PESQUISA APLICADA DEFINIÇÃO BÁSICA DE PESQUISA APLICADA. 2019. Disponível em: <https://www.ifpa.edu.br/documentos-institucionais/0000/3056-anexo-ii-roteiro/file> Acesso em: 16 nov 2021.

MIND BENDING. A História do Python. 2014. Disponível em: <http://mindbending.org/pt/a-historia-do-python> Acesso em: 16 nov 2021.

PISA, P. O que é e como usar o *MySQL*? Artigos. *TechTudo*. 2012. Disponível em: <https://www.techtudo.com.br/noticias/2012/04/o-que-e-e-como-usar-o-mysql.ghtml> Acesso em: 02 out. 2021.

UNIVASF. Como usar o banco de dados SQLite. 2020. Disponível em: <http://www.univasf.edu.br/~brauliro.leal/download/ComoUsarSQLite.pdf>. Acesso em 2 out 2021.