

DESENVOLVIMENTO DE INTERFACE EM PYTHON PARA USO DOS FUNCIONÁRIOS

*Ser acessível é abrir portas para um mundo onde todos têm igualdade de oportunidades,
independentemente das suas diferenças.*

Equipe:

Cauã Ferrigoli Loureiro – RM 553093

Claudio Silva Bispo – RM 553472

Sara Ingrid da Silva Pereira – RM 554021

Sumário

Introdução.....	3
Contexto	3
Desafio	3
Objetivo Geral	4
Objetivo Específico	5
Objetivo para matéria.....	5
Análise e diagnóstico.....	6
Script em Python:.....	8
Tela de teste	13
Conclusão.....	17

Introdução

Imagine um mundo digital onde todos têm acesso fácil e igualitário à informação, independentemente de suas necessidades. Este é o visionário projeto que estamos empenhados em tornar realidade. Como acessar um site e poder escolher o tema? Como ler as informações contidas sem ter que dar um zoom na tela pois a fonte está pequena ou com uma cor que dificulta leitura e entendimento. Já pensou em fazer uma busca usando áudio ao invés de texto?

Nosso projeto tem o objetivo de criar facilidade, simplicidade e incluir suportes que vão ajudar a qualquer tipo de pessoa a localizar um conteúdo com facilidade e sem burocracia.

Contexto

No início, identificamos uma carência significativa na acessibilidade de muitos elementos existentes no site em que estamos estudando e criando melhorias. Pessoas com diferentes habilidades e necessidades estavam sendo abandonadas de diversas maneiras, até mesmo pessoas sem deficiência, que não conhece sobre os produtos da companhia, tinham dificuldades para localizar informações de formas simples e sem burocracia. Imagina que toda vez que precisar entrar em contato com alguém ou ter que assistir a um vídeo, ter que preencher um formulário? Imagine ter que preencher um formulário a cada interação? Ou navegar em várias abas para localizar uma determinada informação e ela estiver no idioma inglês e não ter uma opção para trocar para um idioma base. Imagine deseja tirar dúvidas de forma dinâmica, apenas fazendo perguntas e desejar respostas rápidas e não ter isso disponível.

Desafio

Desenvolver um site que fosse verdadeiramente inclusivo e que consiga atrair o máximo de pessoas possíveis, criando leads para novos negócios. Tínhamos que pensar em cada detalhe, desde o esquema de cores até a funcionalidade de leitura por voz, para que ninguém ficasse de fora e que atenda a todos os requisitos e necessidades. Tornar um site mais inclusivo para pessoas com ou sem deficiência.

Objetivo Geral

1. Investimos tempo considerável em pesquisas e testes para garantir que nosso site atenda a todas as necessidades. Queremos implementar uma variedade de recursos inovadores, sendo eles:
2. Esquema de Cores Personalizável: Permitimos aos usuários escolher entre temas de tela clara ou escura, garantindo uma experiência confortável para todos.
3. Fontes Adaptáveis e Tamanhos Personalizáveis: Desenvolver uma interface que permite ajustar a fonte e seu tamanho, garantindo que a leitura seja fácil para todos. Aqui será criado uma opção visível no navegador onde o usuário vai conseguir configurar a fonte que deseja aplicar no site e tamanho, a cada acesso que fizer, esses padrões já estarão salvos e será aplicado de forma automática no site.
4. Aceite em vários tipos de tecnologias como celular com visualização na horizontal ou vertical, uso em tablets, computadores, Laptops e até mesmo TV.
5. Navegação no site apenas usando o teclado, sem a necessidade de um mouse. Teremos funcionalidades com foco que será acessado através da tecla TAB.
6. Leitura de Tela: Implementar suporte total ou parcial para leitores de tela, permitindo que usuários com deficiência visual acessem e naveguem em nosso site de maneira eficaz.
7. Pesquisa por Texto e Voz: Criação de uma funcionalidade de pesquisa robusta que aceita entrada de texto e voz, proporcionando uma experiência intuitiva para todos os usuários. Essa pesquisa será possível para navegação no site, para realizar busca na opção de busca e também no chatbot.
8. Assistência via chatbot: Introduzir um chatbot inovador que permite aos usuários obter respostas às suas perguntas, pesquisando por texto ou através de comandos de voz. Ele terá a opção de tirar suas dúvidas sobre os produtos e suas características, e falar com um consultor quando desejar.
9. Desenvolver uma funcionalidade de login simplificada, permitindo acesso através das contas do G-mail, LinkedIn, GitHub, Outlook, Apple, e outras plataformas renomadas. O usuário vai poder definir se fará login usando uma plataforma já existente ou se cadastrar direto no site.
10. Implementar formulários de contato mais intuitivos, capazes de serem preenchidos tanto por texto quanto por voz, proporcionando uma experiência inclusiva e eficiente.
11. Utilizar os dados de login simplificado para facilitar o preenchimento dos formulários de contato e para mapear a jornada do visitante pelo site. Isso nos ajuda a compreender suas percepções e necessidades de forma mais precisa.
12. Para os colaboradores, iremos oferecer um painel de consulta ao banco de dados, intuitivo e de fácil utilização. Isso permite que entendam o que o cliente procurou e quanto tempo passou navegando em nosso site. Também fornecemos insights sobre as interações no chat e a atualização de conteúdos, incentivando uma resposta ágil às necessidades dos usuários.
13. Imaginamos um ambiente onde os vídeos possuam legendas incorporadas ou disponíveis separadamente, fornecendo uma descrição abrangente do conteúdo. Isso proporciona a

oportunidade para aqueles que não podem assistir, de ler e absorver o conteúdo de forma igualmente eficaz.

14. Os títulos, subtítulos, textos das páginas precisam ser fáceis de entender e intuitivo.
15. A opção de seleção de idioma para a apresentação dos conteúdos é claramente visível, permitindo que qualquer pessoa escolha como prefere consumir informações - seja por leitura ou audição - em seu idioma nativo ou no que está aprendendo.
16. Opção de definição do contraste da tela, seja para um ambiente Dark ou Light.
17. Vídeos sem play automáticos, o usuário quem vai determinar se vai assistir ou não. Caso não queira assistir, vai apenas ler a descrição fora dos vídeos.
18. Descrição em imagens. Todos os usuários precisam entender do que se refere a imagem e precisa trazer significado.
19. Uma página com a documentação de acessibilidade: Será explicado como os recursos de navegação funcionam e como o usuário pode aplicar durante sua experiência.
20. Criação de um canal para suporte e feedback sobre a experiência no site. Os usuários vão poder comunicar problemas existentes durante a navegação ou avaliar as funcionalidades.
21. Navegação por menus mais simples e acessível. Para uso com ou sem teclado.

Objetivo Específico

1. Site mais acessível de todas as formas possíveis e para qualquer pessoa e em qualquer navegador/dispositivo.
2. Criação de um chatbot inteligente e que forneça todo o suporte para o usuário, sem a necessidade de ficar pesquisando no site ou até mesmo falar com uma pessoa para tirar dúvidas simples sobre produtos, características e informações gerais.

Objetivo para matéria

Estamos em processo de desenvolvimento de uma interface que permitirá aos usuários (funcionários) acessar e consultar os bancos de dados sem a necessidade de conhecimento em programação ou criação de consultas. Esta interface abrangerá informações essenciais para o funcionamento do CHATBOT, como descrições de produtos e dados previamente disponíveis em mais de 1700 perguntas frequentes. Além disso, possibilitará a consulta de LEADS para novas oportunidades de negócio, provenientes de logins feitos por visitantes, clientes e preenchimento de formulários.

Para atingir esse objetivo, estamos integrando as linguagens Python e Java para criar uma interface que conectará o front-end e back-end, permitindo consultas diretas aos bancos de dados e conexões através de APIs. Com isso, buscamos modernizar o chatbot e aprimorar a dinâmica e performance do site, tornando a experiência do usuário mais ágil.

Uma das funcionalidades que planejamos oferecer é a possibilidade de indicar ou solicitar modificações no conteúdo HTML. Essas modificações visam destacar informações relevantes para os

clientes, como a divulgação de produtos específicos e suas características, na página de listagem de produtos. Estamos empenhados em aprimorar a interação e a experiência do usuário no nosso site, tornando-o mais intuitivo e informativo.

Análise e diagnóstico

Decomposição:

Identificamos cada elemento como página inicial, produtos, serviços, suporte, e área de contato. Foi quebrado o problema em partes como por exemplo:

1. O que será necessário manter na página principal? e o que vamos retirar.
2. Como será feito o novo menu? e o que vamos fazer com as informações que acreditamos que estão poluindo o site. Neste momento não vamos descartar as informações, vamos incluir as sessões necessárias em algumas páginas e pegar essas informações que não são necessárias ter no site diretamente e inserir essas informações no chatbot e aproveitar para ensinar o CHATBOT a tirar dúvidas dos clientes em suas pesquisas.

Obs.: Hoje o objetivo do CHATBOT é somente coletar informações do usuário, depois propor um formulário para que ele faça o preenchimento e depois direciona ele para o atendimento humano, o que muitas vezes demorou ou fez a página cair/erro.

Foi considerado os aspectos de acessibilidade de cada componente como: modelo de visualização em tela clara ou escura, aplicação de um modelo que leia o texto completo da página para o cliente ou ele ao clicar na opção específica, cada vez que encostar em uma letra ou frase, será lido para ele. Melhorar aspectos das fontes, deixando algumas maiores, outras com fontes visíveis, mudar o tom da cor para que foque onde precise, mas que não dificulte a leitura de qualquer usuário. Navegação mais intuitiva, onde as imagens terão pequenas legendas para mostrar do que se trata, ao encostar nela, será mostrado o texto do significado para que todos consigam ler e entender. Facilidade de procurar alguma informação com o uso do CHATBOT, seja para procurar por produtos ou saber mais informações sobre alguma característica. Também mostrar um link para direcioná-lo para uma aba correta, sem precisar ficar procurando ou clicando. Por último iremos aplicar a API do Chat-GPT para ampliar o modelo de aprendizado e de passar melhores informações para o usuário, podendo assim criar um lead para um novo negócio.

Reconhecimento de Padrão:

Identificamos padrões de design e usabilidade de cada elemento mencionado.

Reconhecer padrões de acessibilidade, como a utilização correta de etiquetas em formulários, preenchimento do formulário, foco de teclado e como pode obter as informações uma vez que não pode digitar, semântica HTML apropriada, textos que estão fora do padrão, tamanho, cor ou que

não simplifique ou ajude na leitura. Ajustar o botão de pesquisa para que o usuário localize algo através de texto ou por voz, também poder realizar pesquisas no chatbot. São alguns dos padrões identificados e com oportunidade de melhorias.

Abstração:

Focar em elementos essenciais para a experiência do usuário, garantindo acessibilidade em diferentes dispositivos, diferentes tipos de pesquisas de informações no site, seja por escrita, fala ou vídeos com legendas (até mesmo gifs).

Algoritmo:

Organizamos as etapas de desenvolvimento de forma lógica, como:

Primeiro: ouvimos qual era a solicitação/problema;

Segundo: fizemos uma varredura em todo o site para aprender sobre a empresa e os produtos.

Terceiro fizemos todo o processo de navegação como se fossemos o cliente para encontrar determinadas informações e aprender sobre elas para aplicar em "nossa empresa".

Quarto: foram feitos testes de usabilidade e acessibilidade e de acordo com isso, definimos que criaríamos uma interface única usando a linguagem Python, onde o funcionário da Sales Force vai conseguir por exemplo consultar os bancos de dados de logins feitos por visitantes, clientes e dos preenchimentos de formulários. Também vai conseguir consultar o conteúdo dos produtos que teremos no site e suas características, podendo indicar (abrir chamado) para que informações sejam alteradas de acordo com as oportunidades. Isso vai gerar os Leads para novos negócios, sem a necessidade de outras áreas entregarem a informação ou ter somente um canal de contato que hoje são os formulários.

Quinto: estudamos as tecnologias para saber quais aplicar conforme o problema raiz;

Sexto: separamos em partes como iríamos construir as soluções e quem irá fazer cada etapa, respeitando o que foi solicitado.

Plano de Ação:

Implementação de script em Python com objetivo de criar uma interface onde contenha um menu e suas funcionalidades serão relevantes para que cada usuário consiga verificar o conteúdo que contém no site e em determinadas páginas poder indicar uma mudança no conteúdo, deixando-o de forma dinâmica a fim de entender qual é o melhor produto naquele determinado período e assim chamar atenção do cliente. Também consultar os bancos de dados disponíveis para localizar possíveis leads e saber o comportamento do usuário/visitante. Não podendo esquecer das conexões via API para deixar o chatbot mais inteligente e com respostas melhoradas a cada pesquisa.

Com esse menu será possível consultar informações no banco de dados de login de visitantes e saber por onde ele passou e quanto tempo ficou analisando uma informação.

No chatbot teremos um banco de dados com informações sobre produtos, serviços e contatos. O usuário vai poder avaliar se as informações estão de acordo com cada momento, se estão simples, e poder comunicar uma alteração ou incluir o que for necessário. Vamos aproveitar para integrar o chat via API no GPT usando a linguagem para deixar o chat mais Smart e com conteúdo mais atualizados, de acordo com o segmento da nossa empresa.

Conforme a navegação do usuário, vamos inserir um modelo de login simples com dados de redes sociais como (Linkedin, Gmail, Github, Outlook, Apple, entre outros) para coletar os dados de contato e inserir essas informações no banco de dados para consulta de Leads e o usuário vai poder consultar em tempo real, sem a necessidade de uma área ou pessoa ter que entregar essa informação para ele. O Objetivo é evitar o preenchimento de formulários e coletar mais rápido os dados.

Objetivo principal é concentrar os dados em um único lugar, tornando dinâmico, simples e rápido para tomar qualquer decisão.

Script em Python:

```
#----- ( Baixar bibliotecas ) -----#
import json
import os
print(os.getcwd())

#----- ( Função para chama o json menu ) -----#
# Função para carregar os dados do arquivo JSON - Lá precisa conter a nossa base de
# busca, onde vou adicionar, excluir, alterar dados
def carregar_dados():
    with open('./menu_data.json', 'r') as arquivo:
        dados = json.load(arquivo)
    return dados

#----- ( Função para salvar dados ) -----#
# Função para salvar os dados de volta no arquivo JSON
def salvar_dados(dados):
    with open('./menu_data.json', 'w') as arquivo:
        json.dump(dados, arquivo, indent=4)

#----- ( Função para voltar ou encerrar o programa ) -----#

def opcao():
    print("Para voltar ao menu anterior, digite 0 ou 1 para encerrar o programa")
    opcao = input("Escolha uma opção: ")
    opcao = int(opcao)
```

```
if opcao == 0:
    print("Você definiu a opção de voltar ao menu principal")
elif opcao == 1:
    print("Encerrando o programa...")
    exit()

else:
    pass

#------( Função para Exibir o menu )-----#

# Função para exibir os menus
def exibir_menus():

    dados = carregar_dados()
    menus = dados.get('menus', [])

    print("### As opções disponíveis hoje! ###")
    for menu in menus:
        print(f"ID: {menu['id']} | Nome: {menu['nome']} | Link: {menu['link']}")

    opcao()

#------( Função para realizar a consulta no banco, através de conexão ao banco
# Estamos imaginando usar o SQL Server Ou Oracle por exemplo )-----#

def conectar_sql():
    server = ''
    database = ''
    user = ''
    password = ''

    tentativas = 3

    for tentativa in range(1, tentativas + 1):
        try:
            conexao = pymssql.connect(server=server, database=database, user=user,
password=password)
            print(f"Conexão bem-sucedida na tentativa {tentativa}")
            return conexao
        except Exception as e:
            print(f"Tentativa {tentativa} falhou. Erro: {e}")
            if tentativa == tentativas:
                raise
```

```
def gerar_query():

    # Query - Aqui pode modificar se precisar
    query = """ Inserir a consulta aqui quando criarmos o banco """

    # Fazer a conexão e olhar os dados
    conexao = conectar_sql()
    df = pd.read_sql_query(query, conexao)

    conexao.close()

    return df

#----- ( Função para consultar o banco de dados - Log de visitantes )-----
-#

# Função para exibir os menus
def consultar_banco_visitantes():

    print("### Acesso ao banco de dados dos visitantes ###")

    # query para consulta do banco de dados. Aqui será onde a consulta vai ocorrer
    # sem a necessidade de montar outras queries

    #df = gerar_query()
    #print(df)

    opcao()

#----- ( Função para consultar o banco de dados de logs de clientes )-----
#

# Função para exibir os menus
def consultar_banco_clientes():

    print("### Acesso ao banco de dados - Logs de clientes ###")

    # query para consulta do banco de dados. Aqui será onde a consulta vai ocorrer
    # sem a necessidade de montar outras queries

    #df = gerar_query()
    #print(df)

    opcao()

#----- ( Função para consultar o banco de dados - Formulários preenchidos )---
-----#
```

```
# Função para exibir os menus
def consultar_banco_formularios():

    print("### Acesso ao banco de dados - Formulários preenchidos ###")

    # query para consulta do banco de dados. Aqui será onde a consulta vai ocorrer
    # sem a necessidade de montar outras queries

    #df = gerar_query()
    #print(df)

    opcao()

#-----( Função para consultar o banco de dados )-----#

# Função para exibir os menus
def consultar_leads():

    print("### Acesso a consulta de Leads ###")

    # query para consulta do banco de dados. Aqui será onde a consulta vai ocorrer
    # sem a necessidade de montar outras queries

    #df = gerar_query()
    #print(df)

    opcao()

#-----( Função para adicionar o menu )-----#

# Função para adicionar um novo menu
def adicionar_conteudo():
    nome = input("Digite o nome do novo menu: ")
    link = input("Digite o link do novo menu: ")

    dados = carregar_dados()
    menus = dados.get('menus', [])

    novo_menu = {'id': len(menus) + 1, 'nome': nome, 'link': link}
    menus.append(novo_menu)

    dados['menus'] = menus
    salvar_dados(dados)

    print(f"Menu '{nome}' adicionado com sucesso!")

    opcao()
```

```
#----- ( Função para remover algum menu )-----#

# Função para remover um menu
def remover_conteudo():
    id_menu = int(input("Digite o ID do menu que deseja remover: "))

    dados = carregar_dados()
    menus = dados.get('menus', [])

    for menu in menus:
        if menu['id'] == id_menu:
            menus.remove(menu)
            dados['menus'] = menus
            salvar_dados(dados)
            print(f"Menu ID {id_menu} removido com sucesso!")
            return

    print(f"Nenhum menu encontrado com ID {id_menu}.")

    opcao()

#----- ( Função para fechar o programa )-----#

# Função para remover um menu
def sair_programa():
    print(f"Programa encerrado.")
    exit()

#----- ( Função para o menu de opções )-----#
# Menu de opções
def menu_principal():
    while True:
        print("\n### Menu de Administração para funcionários ###")
        print("1. Exibir Menus do site")
        print("2. Adicionar conteúdo")
        print("3. Remover conteúdo")
        print("4. Consultar Banco de dados logs - visitantes")
        print("5. Consultar Banco de dados logs - clientes")
        print("6. Consultar Banco de dados formulário")
        print("7. Consultar Leads")
        print("8. Sair")

        opcao = input("Escolha uma opção: ")

        if opcao == "1":
            exibir_menus()
        elif opcao == "2":
```

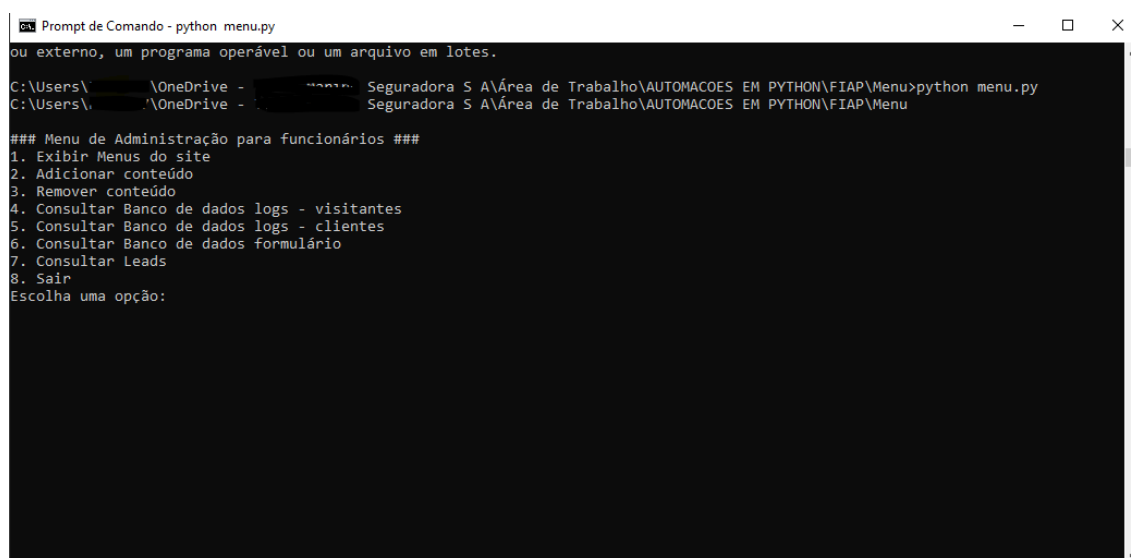
```
        adicionar_conteudo()
    elif opcao == "3":
        remover_conteudo()
    elif opcao == "4":
        consultar_banco_visitantes()
    elif opcao == "5":
        consultar_banco_clientes()
    elif opcao == "6":
        consultar_banco_formularios()
    elif opcao == "7":
        consultar_leads()
    elif opcao == "8":
        sair_programa()
        break
    else:
        print("Opção inválida. Tente novamente.")

#------( Função para chamar as demais funções )-----#

# Iniciar o menu principal - Aqui vou rodar todos os processos
if __name__ == "__main__":
    menu_principal()
```

Tela de teste

1. Exibir menu inicial



```

Prompt de Comando - python menu.py
ou externo, um programa operável ou um arquivo em lotes.

C:\Users\... \OneDrive - ... Seguradora S A\Área de Trabalho\AUTOMACOES EM PYTHON\FIAP\Menu>python menu.py
C:\Users\... \OneDrive - ... Seguradora S A\Área de Trabalho\AUTOMACOES EM PYTHON\FIAP\Menu

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção:
```

2. Ao clicar no menu inicial, ele vai mostrar as opções que temos disponíveis no site hoje e também duas opções ao usuário, onde uma será para voltar a tela anterior e outra para encerrar a aplicação.

```
### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 1
### As opções disponíveis hoje! ###
ID: 1 | Nome: Consultar Produtos | Link: https://www.salesforce.com/br/products/sales-cloud/pricing/
ID: 2 | Nome: Para Industrias | Link: https://www.salesforce.com/br/solutions/industries/automotive/overview/
ID: 3 | Nome: Sobre Ensino | Link: https://trailhead.salesforce.com/pt-BR
ID: 4 | Nome: Suporte | Link: https://help.salesforce.com/s/?language=en_US
ID: 5 | Nome: Para Empresa | Link: https://www.salesforce.com/br/company/our-story/
ID: 6 | Nome: Consultar Banco de Dados | Link: Inserir a API do banco de dados aqui
ID: 7 | Nome: Consultar Leads | Link: Inserir a API do banco de dados aqui - Que contenha os principais clientes para contato
ID: 8 | Nome: clear | Link: exit
ID: 9 | Nome: exit | Link: exit
ID: 10 | Nome: teste | Link: teste.com
ID: 11 | Nome: 1 | Link: 1
ID: 12 | Nome: 1 | Link: 1
Para voltar ao menu anterior, digite 0 ou 1 para encerrar o programa
Escolha uma opção:
```

- Na opção 2, ele será capaz de informar uma opção do menu que quer incluir/adicionar e após isso, voltar para o menu principal ou fechar a aplicação.

```

CA: Prompt de Comando - python menu.py

Você definiu a opção de voltar ao menu principal

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 2
Digite o nome do novo menu:

```

```

CA: Prompt de Comando - python menu.py

Você definiu a opção de voltar ao menu principal

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 2
Digite o nome do novo menu: teste
Digite o link do novo menu: google.com.br
Menu 'teste' adicionado com sucesso!
Para voltar ao menu anterior, digite 0 ou 1 para encerrar o programa
Escolha uma opção:

```

- Na opção 3, ele será capaz de indicar uma exclusão, assim como as opções de voltar e encerrar a aplicação.

```

CA: Prompt de Comando - python menu.py

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 3
Digite o ID do menu que deseja remover:

```



```

Prompt de Comando - python menu.py

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 3
Digite o ID do menu que deseja remover: 5
Menu ID 5 removido com sucesso!

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção:

```

5. Opção 4,5 e 6 será para realizar consulta ao bando de dados de logins feitos pelos visitantes, clientes e preenchimentos de formulários, onde teremos a data, hora, e-mail, nome e o que ele pesquisou. Assim gerando um possível lead para novos negócios. Inserimos a opção de voltar e encerrar a aplicação.

```

Prompt de Comando - python menu.py

### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 4
### Acesso ao banco de dados dos visitantes ###
Para voltar ao menu anterior, digite 0 ou 1 para encerrar o programa
Escolha uma opção:

```

6. A opção 7 é uma possível integração ao banco de dados da empresa, onde contenham informações de clientes para contatar, de acordo com o cadastro do chat, dos consultores do atendimento, até mesmo alguma base de dados existentes que eles tenham criado.
7. Última opção é a 8 que é para encerrar a aplicação.

CIT Prompt de Comando

```
### Menu de Administração para funcionários ###
1. Exibir Menus do site
2. Adicionar conteúdo
3. Remover conteúdo
4. Consultar Banco de dados logs - visitantes
5. Consultar Banco de dados logs - clientes
6. Consultar Banco de dados formulário
7. Consultar Leads
8. Sair
Escolha uma opção: 8
Programa encerrado.
```

Conclusão

Nossa dedicação à acessibilidade é contínua e vai além. Continuaremos a aprimorar nossa proposta de site e a explorar novas formas de tornar a web um ambiente verdadeiramente inclusivo, em total alinhamento com os objetivos da nossa empresa.

A implementação de um chatbot ainda mais avançado reforça nossa abordagem inovadora. Ao empregar tecnologias de processamento de linguagem natural de última geração, nosso chatbot proporciona respostas precisas e relevantes em tempo real, oferecendo suporte eficaz e personalizado aos visitantes do site. Com essa integração, as empresas desfrutam de ganhos significativos. Têm a capacidade de criar um ambiente digital que promove inclusão, aprimorar a eficiência operacional e proporcionar experiências excepcionais aos clientes.

Além disso, a combinação de um site acessível e um chatbot tecnologicamente avançado simplifica a captação de leads, tornando o processo mais intuitivo e eficaz. A interação personalizada e a facilidade de acesso contribuem para uma maior geração de leads qualificados, impulsionando o crescimento e a competitividade no mercado.

Ao alinharmos nossa visão com os princípios da Salesforce, Sociedade 5.0 e 4.0, não apenas incorporamos tecnologias avançadas como IA e automação, uso da linguagem Python, mas também capacitamos as empresas/clientes a proporcionarem experiências mais ricas e personalizadas aos seus clientes e funcionários. Ao criar um site extremamente acessível, reforçamos nosso compromisso com a inclusão, tornando a experiência online acessível a todos, independentemente de suas necessidades e habilidades.