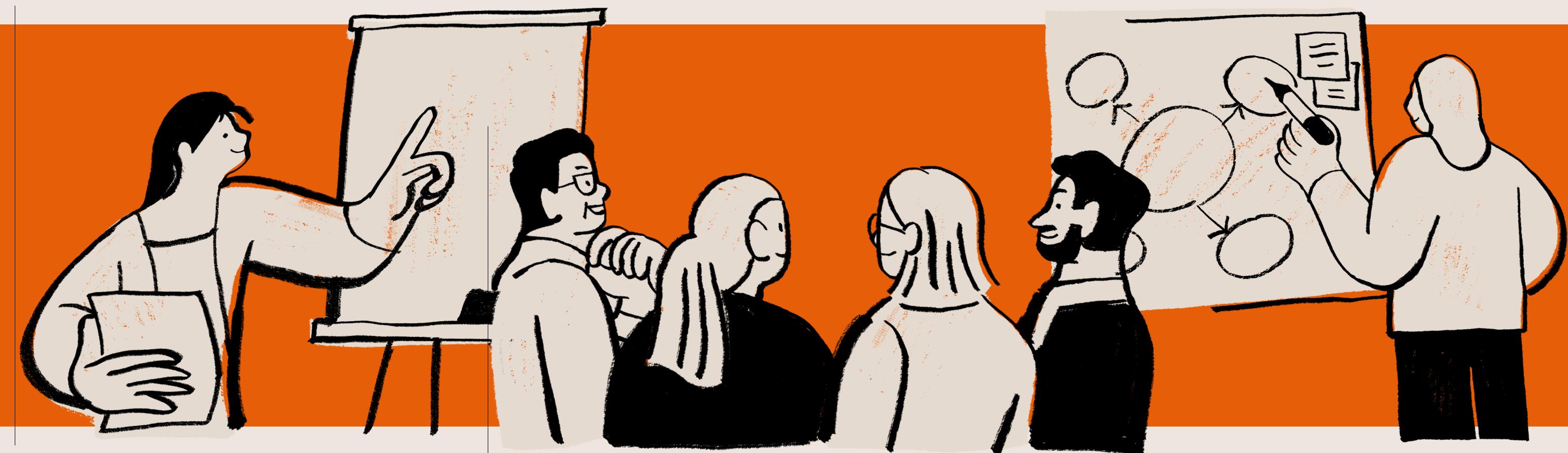


ESTACIO - PROJETO EM PYTHON



Entendimento do Cliente e do Problema

O cliente deste projeto era uma amiga de trabalho de um dos membros do grupo, que enfrentava dificuldades para:

- Criar senhas fortes
- Manter uma organização segura no dia a dia digital

Essa necessidade constante de criar novas senhas e não conseguir guardá-las corretamente gerava insegurança e perda de tempo.



Objetivo do Projeto:

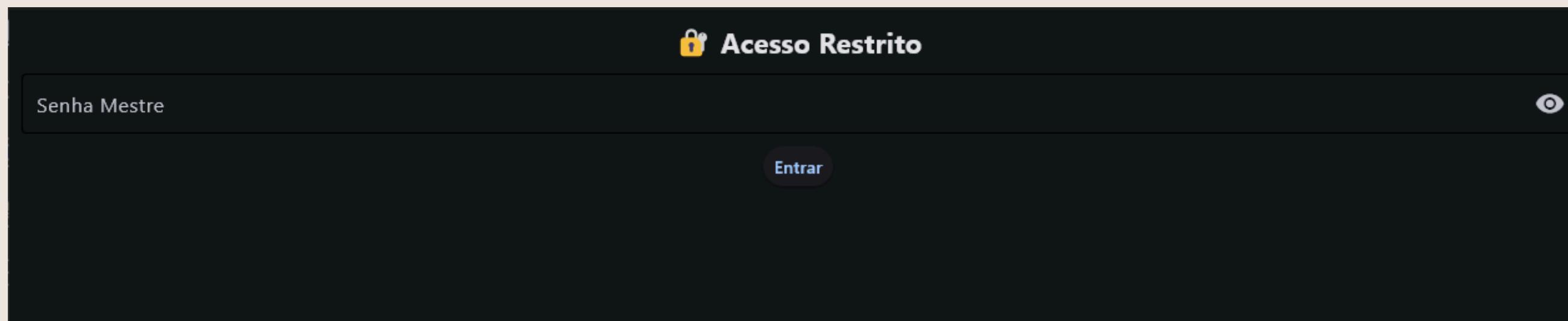
Desenvolver uma ferramenta simples, rápida e segura para gerar senhas aleatórias, fortes e personalizadas, facilitando a vida da cliente e aumentando a segurança das suas contas.

Gerador de Senhas com Login (Python + Flet)

Este projeto cria um sistema simples e seguro para gerar senhas personalizadas.

Inclui:

- Tela de login com senha mestre
- Interface moderna usando Flet
- Opções avançadas para gerar senhas
- Função para copiar facilmente



Estrutura do Código

Nesta etapa, foi desenvolvida a base do programa:

- Importação das bibliotecas (flet, random, string)
- Definição da senha mestre
- Estrutura principal main() responsável pela janela

```
1 import flet as ft
2 import random
3 import string
4
5 # Senha mestre usada para liberar o acesso ao programa
6 MASTER_PASSWORD = "admin123"
```



Explicação:

Flet controla a interface, enquanto random e string fornecem os caracteres usados na senha.

Tela de Login

A função `login_screen()` cria:

- Campo de senha com opção de revelar
- Botão “Entrar”
- Validação da senha
- Mensagem de erro em caso de tentativa incorreta

```
16  # ----- TELA DE LOGIN -----
17  def login_screen():
18      # Campo para digitar a senha (oculto por segurança)
19      senha_input = ft.TextField(
20          password=True,
21          can_reveal_password=True, # permite mostrar/ocultar
22          label="Senha Mestre"
23      )
24
25
26      # Texto de erro caso a senha esteja incorreta
27      erro_text = ft.Text("", color="red")
28
29      # Função que valida a senha digitada
30      def validar(e):
31          if senha_input.value == MASTER_PASSWORD:
32              main_screen() # abre a tela principal se acertar
33          else:
34              erro_text.value = "Senha incorreta!"
35              page.update()
36
37      # Botão de login
38      btn_login = ft.ElevatedButton("Entrar", on_click=validar)
```



Explicação:

O programa só libera o acesso à tela principal após digitar a senha mestre.
Se estiver errada → mostra mensagem de erro.

Tela Principal (Gerador)

A função `main_screen()` monta toda a interface de geração de senhas:

- Slider para escolher o tamanho
- Checkboxes para selecionar tipos de caracteres
- Campo para exibir a senha
- Botão de copiar senha

```
54
55     # ----- TELA PRINCIPAL -----
56     def main_screen():
57         # Slider para escolher o tamanho da senha
58         tamanho_slider = ft.Slider(
59             min=1, max=20, divisions=19, value=12,
60             label="Tamanho: {value}"
61         )
62
63         # Checkboxes para escolher o que a senha deve conter
64         chk_maius = ft.Checkbox(label="Letras maiúsculas (A-Z)", value=True)
65         chk_minus = ft.Checkbox(label="Letras minúsculas (a-z)", value=True)
66         chk_num = ft.Checkbox(label="Números (0-9)", value=True)
67         chk_esp = ft.Checkbox(label="Caracteres especiais (!, @, #, ...)", value=True)
68
69         # Campo onde a senha final será exibida
70         senha_output = ft.TextField(label="Senha Gerada", read_only=True)
71
```



Explicação:

A interface é simples e direta, permitindo personalização total da senha.

Lógica de Geração da Senha

O processo é feito pela função `gerar_senha()`:

- Coleta os tipos de caracteres selecionados
- Monta uma lista de possibilidades
- Gera a senha com `random.choice()`
- Exibe no campo final

```
71
72     # Função que gera a senha
73     def gerar_senha(e):
74         chars = "" # conjunto de caracteres que serão usados
75
76         # Adiciona tipos de caracteres conforme os checkboxes marcados
77         if chk_maius.value:
78             chars += string.ascii_uppercase
79         if chk_minus.value:
80             chars += string.ascii_lowercase
81         if chk_num.value:
82             chars += string.digits
83         if chk_esp.value:
84             chars += "!@#$%&*()_-+={}[]<>?"
85
86         # Se nenhum tipo foi selecionado, avisa o usuário
87         if chars == "":
88             senha_output.value = "Selecione ao menos 1 tipo!"
89             page.update()
90             return
```



Explicação:

A senha é criada de forma totalmente aleatória e segura.

Função de Copiar Senha

A função copiar_senha():

- Copia a senha gerada para a área de transferência
- Exibe uma notificação no app (SnackBar)

```
99  
00      # Função para copiar a senha para a área de transferência  
01  def copiar_senha(e):  
02      if senha_output.value:  
03          page.set_clipboard(senha_output.value) # copia  
04          page.snack_bar = ft.SnackBar(ft.Text("Senha copiada!"))  
05          page.snack_bar.open = True  
06          page.update()  
07
```



Explicação:

O usuário copia a senha com apenas um clique, facilitando o uso.

Tempo de Desenvolvimento (1 Semana)

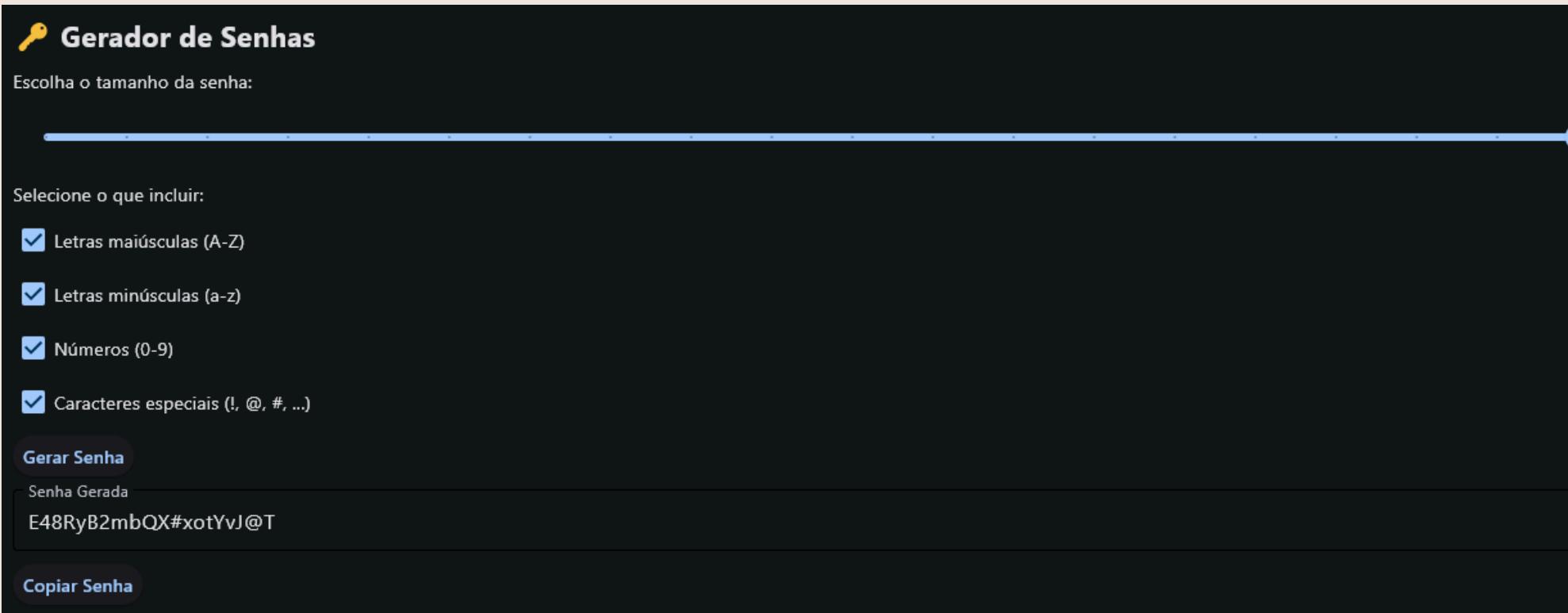
Linha do tempo do projeto:

- Dia 1-2: Planejamento da interface e lógica
- Dia 3: Construção da tela de login
- Dia 4: Montagem da tela principal
- Dia 5: Lógica da geração de senhas
- Dia 6: Testes, ajustes e refinamentos
- Dia 7: Revisão final

Conclusão

O projeto demonstra:

- Uso eficiente de Python com Flet
- Criação de interfaces funcionais
- Modularidade (cada parte em sua função)
- Resultado simples, seguro e totalmente funcional



O sistema atingiu o objetivo principal:

✓ gerar senhas de forma prática, rápida e personalizável.

MEMBROS

Luis Victor Gonçalves de Oliveira Santos
Cauã Moreira de Souza

REFERÊNCIAS BIBLIOGRÁFICAS

Copilot

DankiCode

W3Scholls

DevAprender

Hashtag Programação

ChatGPT

Estacio



**OBRIGADO
PELA ATENÇÃO!!!**