



LISTEN: AVALIADOR DE ÁLBUNS E SINGLES

TEMA: RECOMENDAÇÕES INTELIGENTES BASEADO NAS AVALIAÇÕES DOS MELHORES ÁLBUNS E SINGLES

João Cândido,
Naielly Moura,
Suelen Silva.
Tópicos Especiais



INTRODUÇÃO



- O Listen é um aplicativo projetado para aprimorar a experiência de quem ama música;
 - plataforma onde os usuários podem avaliar álbuns e singles, proporcionando feedback detalhado sobre suas preferências musicais;
 - Com nas avaliações, o aplicativo utilizará algoritmos de recomendação para sugerir novos artistas, álbuns e músicas alinhados aos gostos do usuário, criando uma experiência musical personalizada.
-

CRONOGRAMA





Cronograma					
TAREFA ID	TAREFA NOME	INTEGRANTE	PREVISÃO TERMINO	PRIORIDADE	STATUS
1	Requisitos funcionais	Suelen	18/09/2024	ALTA	FINALIZADA
2	Requisitos não funcionais	Naielly	18/09/2024	ALTA	FINALIZADA
3	Casos de teste	Naielly	25/09/2024	ALTA	FINALIZADA
4	Casos de teste	Suelen	25/09/2024	ALTA	FINALIZADA
5	Introdução(Colocar começo, meio e fim)	Suelen	02/10/2024	MÉDIA	FINALIZADA
6	Colocar titulo nas tabelas de casos de teste	João	02/10/2024	MÉDIA	FINALIZADA
7	validar_email(Criar class e TDD)	Naielly	05/10/2024	ALTA	FINALIZADA
8	validar_campos_obrigatorios(Criar class e TDD)	Suelen	05/10/2024	ALTA	FINALIZADA
9	validar_senhas(Criar class e TDD)	João	05/10/2024	ALTA	FINALIZADA
10	Inserir dados de login válidos(Teste de interface)	João	09/10/2024	ALTA	FINALIZADA
11	Inserir e recuperar dados de usuários(Teste de interface)	João	09/10/2024	ALTA	FINALIZADA
12	Apresentação do projeto	Todos	16/10/2024	ALTA	EM ANDAMENTO





Tópicos especiais

Bem-vindo (ou vinda) ao começo do grupo Tópicos especiais.

 **naielly** iniciou uma chamada. Hoje às 19:23

 **João Cândido** Hoje às 19:29
Oi, boa noite


 **Suelen_Vitoria** Hoje às 19:30
arrumei a introdução da documentação e coloquei referências
boa noite gente


 **João Cândido** Hoje às 19:57

Web renderer

You can change default "cavaskit" web renderer ([more about renderers here](#)) to "html" with `--web-renderer` option:


```
flutter build web --web-renderer html
```


 **naielly** Hoje às 19:58
`ft.app(target=main, view=ft.WEB_BROWSER)`


 **João Cândido** Hoje às 19:58

- **HTML renderer** - uses a combination of HTML elements, CSS, Canvas elements, and SVG elements. This renderer has a smaller download size.
- **CanvasKit renderer** - this renderer is fully consistent with Flutter mobile and desktop, has faster performance with higher widget density, but adds about 2MB in download size.

`ft.app(main, view=ft.AppView.WEB_BROWSER, web_renderer=ft.WebRenderer.HTML)`

 **naielly** Hoje às 20:01
<html> element must have a lang attribute: The <html> element does not have a lang attribute

 **Suelen_Vitoria** Hoje às 20:08
https://www.canva.com/design/DAGTr_Ouk4k/AfYentRM0FnJlqb8fPXbQQ/edit?utm_content=DAGTr_Ouk4k&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
link dos slides

 **naielly** Hoje às 20:10

Cronograma						
TAREFA ID	TAREFA NOME	INTEGRANTE	PREVISÃO	TERMINO	PRIORIDADE	STATUS

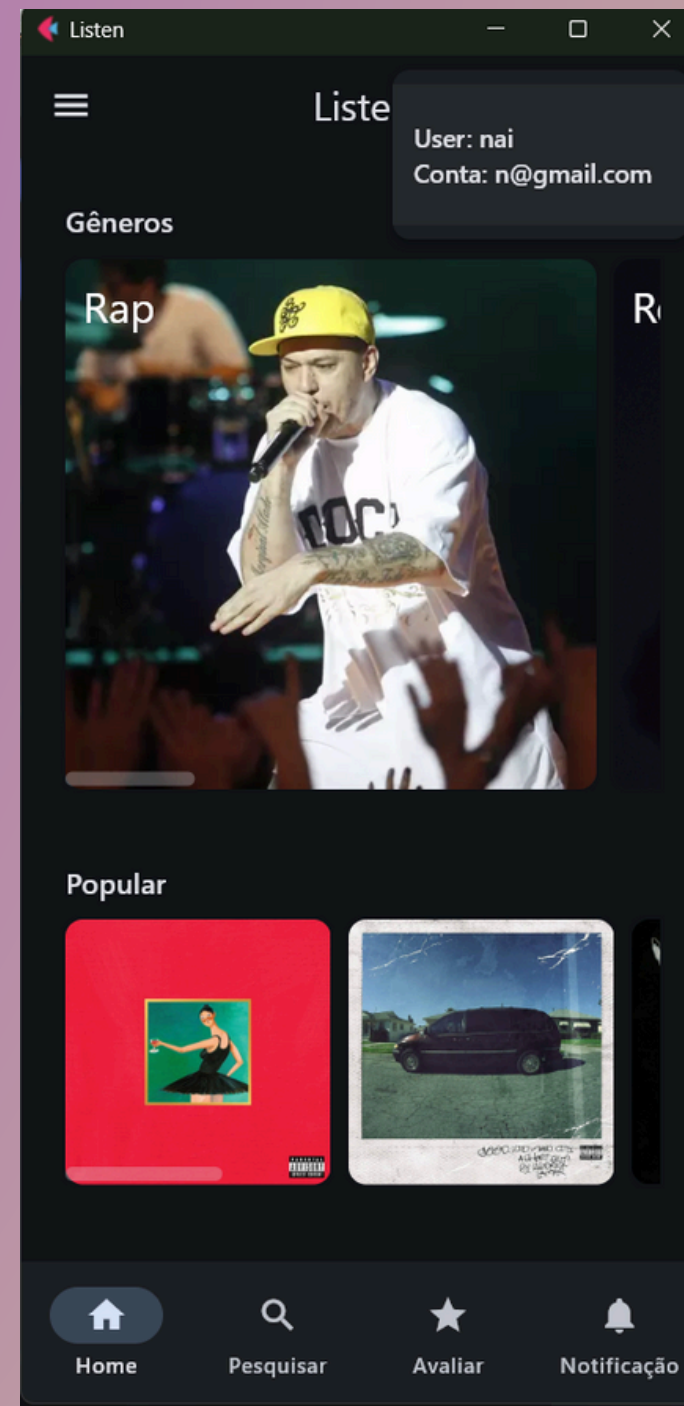
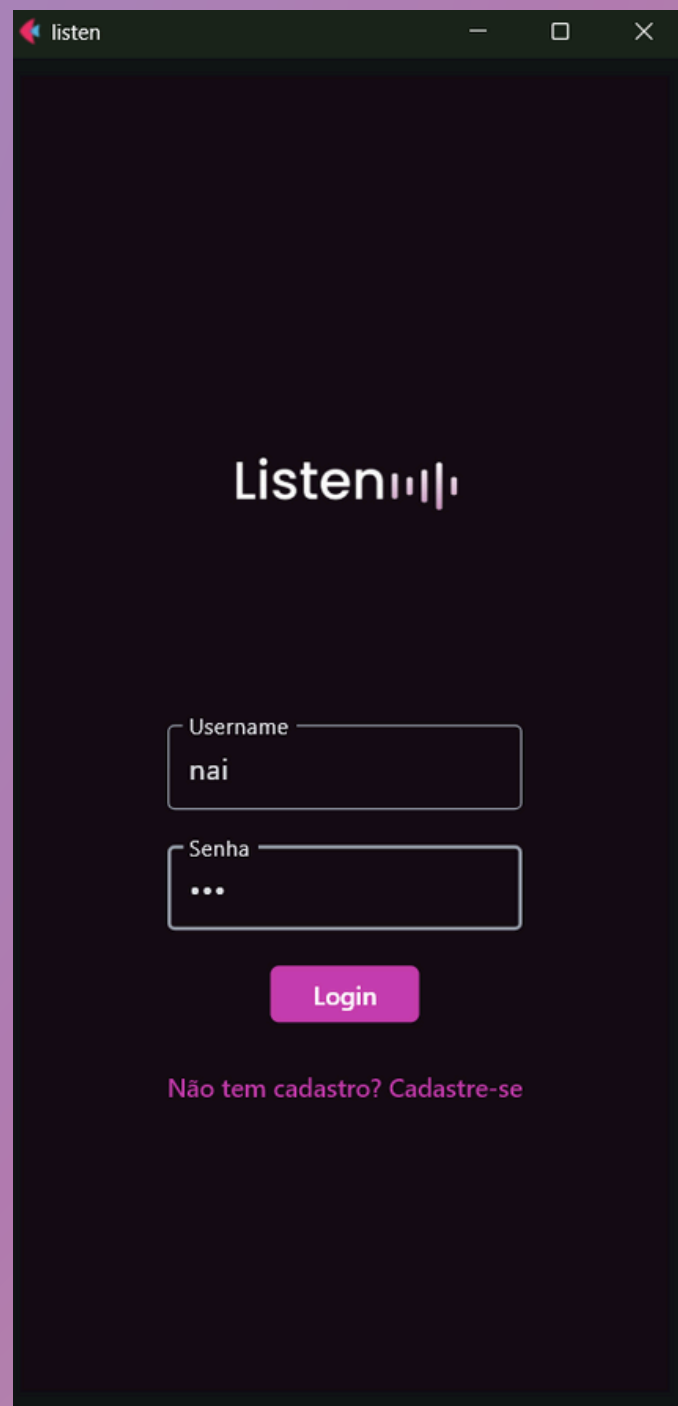
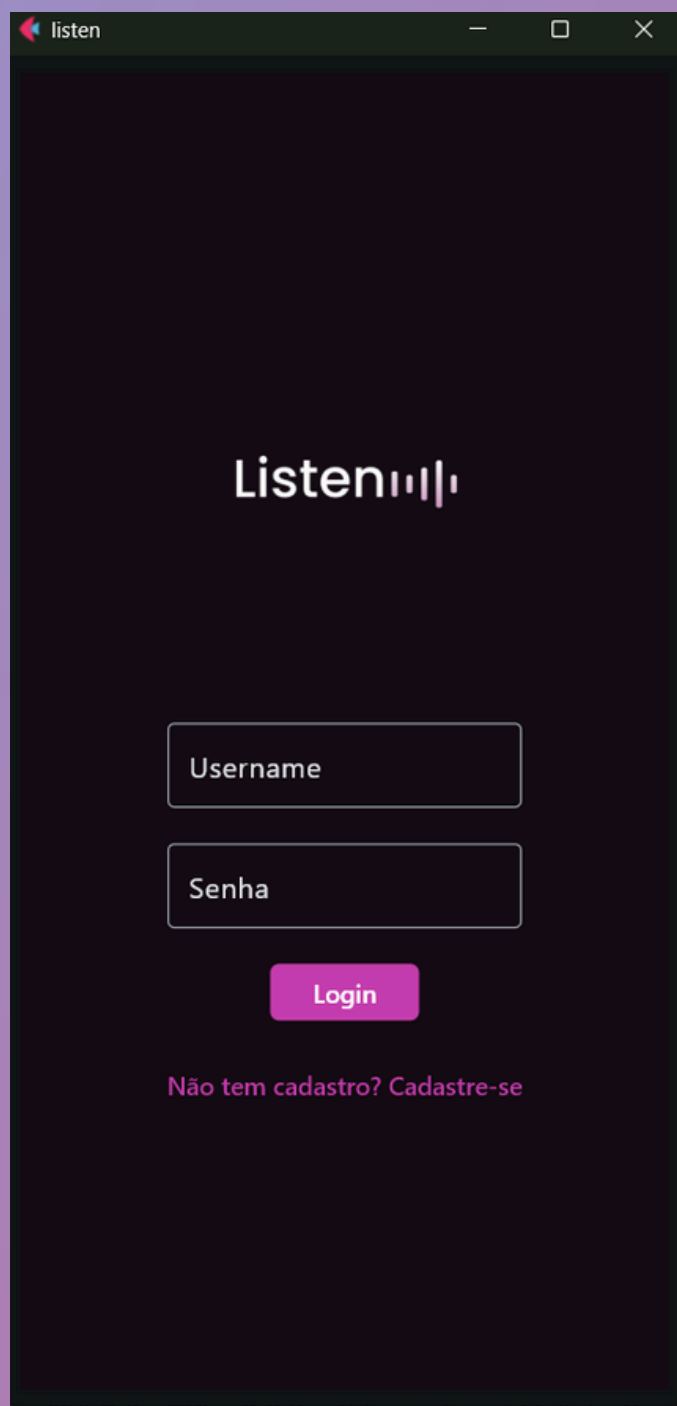
NOVO

TESTES



Caso de teste	CT - 3.1.3 - Inserir dados de login válidos
Pré-condições	Estar na tela de login
Procedimento	1) O usuário informa um email e senha cadastrados no sistema. 2) O sistema verifica se o usuário está cadastrado e se a senha está correta. 3) O sistema permite o acesso e redireciona o usuário para a página principal.
Resultado esperado	Direcionar para a página principal
Dados de entrada	Login e senha válidos
Prioridade	Alta
Ambiente	Windows 10/11
Técnica	Manual
Iteração	1º Iteração

TESTES

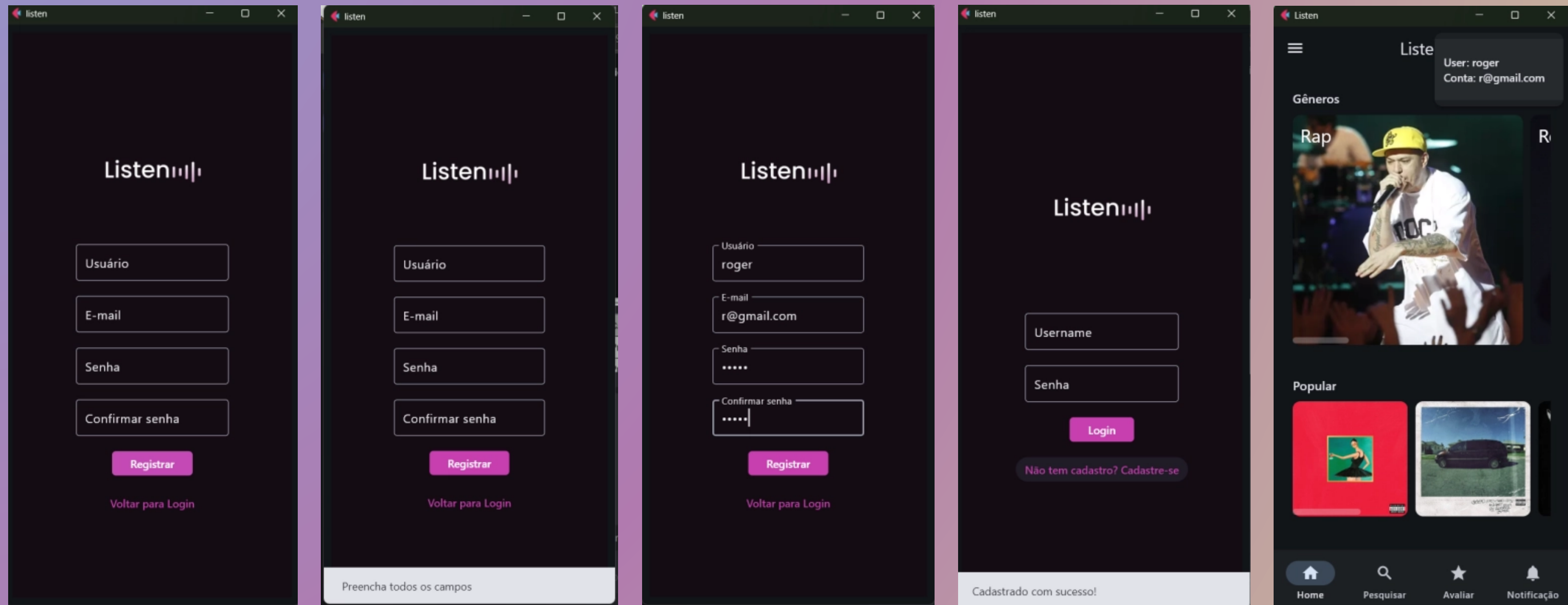


TESTES



Caso de teste	CT - 3.5.1 Inserir e recuperar dados de usuários
Pré-condições	Estar na tela de cadastro de usuários
Procedimento	<ol style="list-style-type: none">1) Acessar a tela de cadastro de usuários.2) Inserir os dados do novo usuário (nome, email, senha).3) Enviar os dados para o sistema.4) Acessar a tela de login.5) Inserir as credenciais cadastradas.6) <u>Verificar</u> se o login é realizado com sucesso e se os dados estão corretos.
Resultado esperado	O usuário deve ser inserido no banco de dados e recuperado corretamente durante o login, permitindo o acesso ao sistema.
Dados de entrada	Nome: "Naielly Moura" Email: "naielly@email.com" Senha: " <u>senha1234</u> "
Prioridade	Alta

TESTES



TESTES



```
main.py  Help  <  >  Listen V
main.py  teste_validar_senha.py  validar_senha.py X
Telas > componentes > validar_senha.py > ...
1  class ValidadorSenhas:
2      def validar(self, senha, confirmar_senha):
3          if senha != confirmar_senha:
4              return "As senhas não coincidem"
5          return None
6
```

```
main.py  teste_validar_senha.py X  validar_senha.py  teste.py  can
Telas > componentes > teste_validar_senha.py > ...
1  import unittest
2  from validar_senha import ValidadorSenhas
3
4  class TestValidadorSenhas(unittest.TestCase):
5      def setUp(self):
6          self.validador = ValidadorSenhas()
7
8      def test_senhas_iguais(self):
9          resultado = self.validador.validar("senha123", "senha123")
10         self.assertIsNone(resultado)
11
12         def test_senhas_diferentes(self):
13             resultado = self.validador.validar("senha123", "senha22")
14             self.assertEqual(resultado, "As senhas não coincidem")
15
16     if __name__ == "__main__":
17         unittest.main()
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TEST RESULTS  PORTS
..
-----
Ran 2 tests in 0.000s
OK
PS C:\Users\naiel\Downloads\Listen Versão Final 25 09\Listen Versão Final 25 09> & "c:\Users\naiel\Downloads\Listen Versão Final 25 09\.venv\Scripts\python.exe" "c:\Users\naiel\Downloads\Listen Versão Final 25 09\teste_validar_senha.py"
..
-----
Ran 2 tests in 0.000s
OK
PS C:\Users\naiel\Downloads\Listen Versão Final 25 09\Listen Versão Final 25 09>
(Listen Versão Final 25 09) Indexing completed
```

TESTES



```
main.py  Help  ← →  Listen Versão Final 25 09

main.py  validar_email.py X
Telas > componentes > validar_email.py > ...
1  import re
2
3  class ValidadorEmail:
4      @staticmethod
5      def validar(email):
6          padrao = r'^[a-zA-Z0-9_+.-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+$'
7          if not re.match(padrao, email):
8              return "E-mail inválido"
9          return None
10
```

```
main.py  teste_validador_email.py X
Telas > componentes > teste_validador_email.py > ...
1  import unittest
2  from validar_email import ValidadorEmail
3
4  class TestValidadorEmail(unittest.TestCase):
5
6      def test_email_valido(self):
7          self.assertIsNone(ValidadorEmail.validar("usuario@dominio.com"))
8
9      def test_email_invalido_sem_aroba(self):
10         self.assertEqual(ValidadorEmail.validar("usuariodominio.com"), "E-mail inválido")
11
12         def test_email_invalido_sem dominio(self):
13             self.assertEqual(ValidadorEmail.validar("usuario@dominio"), "E-mail inválido")
14
15         def test_email_invalido_caracteres_especiais(self):
16             self.assertEqual(ValidadorEmail.validar("usuario@dominio!com"), "E-mail inválido")
17
18     if __name__ == "__main__":
19         unittest.main()
20
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TEST RESULTS  PORTS

enha.py"
..
-----
Ran 2 tests in 0.000s

OK
PS C:\Users\naiel\Downloads\Listen Versão Final 25 09\Listen Versão Final 25 09> & "c:/Users/naiel/Downloads/Li
Final 25 09/.venv/Scripts/python.exe" "c:/Users/naiel/Downloads/Listen Versão Final 25 09/Listen Versão Final 2
_email.py"
....
-----
Ran 4 tests in 0.001s

OK
PS C:\Users\naiel\Downloads\Listen Versão Final 25 09\Listen Versão Final 25 09>
a (Listen Versão Final 25 09) ... Indexing completed ... In 3 Col
```

TESTES



```
final Help  <  >  Listen Versão Final 25 09

main.py  validar_campos_obrigatorios.py X

Telas > componentes > validar_campos_obrigatorios.py > ValidadorCamposObrigatorios

1 class ValidadorCamposObrigatorios:
2     def validar(self, usuario, email, senha, confirmar_senha):
3         if not usuario or not email or not senha or not confirmar_senha:
4             return "Preencha todos os campos"
5         return None
6
```

```
main.py  teste_validar_campos_obrigatorios.py X

Telas > componentes > teste_validar_campos_obrigatorios.py > TestValidadorCamposObrigatorios > setUp

1 import unittest
2 from validar_campos_obrigatorios import ValidadorCamposObrigatorios
3
4 class TestValidadorCamposObrigatorios(unittest.TestCase):
5     def setUp(self):
6         self.validador = ValidadorCamposObrigatorios()
7
8     def test_campos_preenchidos(self):
9         resultado = self.validador.validar("jose marcos", "email@example.com", "senha123", "senha123")
10        self.assertIsNone(resultado)
11
12    def test_campos_vazios(self):
13        resultado = self.validador.validar("", "email@example.com", "senha123", "senha123")
14        self.assertEqual(resultado, "Preencha todos os campos")
15
16 if __name__ == "__main__":
17     unittest.main()

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TEST RESULTS  PORTS

_email.py"
....
-----
Ran 4 tests in 0.001s

OK
PS C:\Users\naiel\Downloads\Listen Versão Final 25 09\Listen Versão Final 25 09> & "c:/Users/naiel/Downloads/Listen Versão Final 25 09/Listen Versão Final 25 09/.venv/Scripts/python.exe" "c:/Users/naiel/Downloads/Listen Versão Final 25 09/Telas/componentes/validar_campos_obrigatorios.py"
..
-----
Ran 2 tests in 0.000s

OK
PS C:\Users\naiel\Downloads\Listen Versão Final 25 09\Listen Versão Final 25 09>
Listen Versão Final 25 09) ... Indexing completed
Ln 6, Col 55  Spaces: 4  UTF-8  CRLF
```