

MOBILE APP COM FLET

▼ Configurações iniciais

Instale o Flet

```
pip install flet
pip install requests
```

Define as propriedades da página:

```
page.title = "Cadastro App"
page.window_width = 400
```

Defina a função a ser chamada em todas as trocas de rota:

```
page.on_route_change = route_change
page.go('/')
```

Chame as páginas certas a cada rota:

```
def route_change(e):
    page.views.clear()
    if page.route == "/":
        home_page()
    elif page.route == '/review':
        review_page()

    page.update()
```

Crie a home_page:

```
def home_page():

    nome_input = ft.TextField(label="Nome do Produto", text_align=ft.TextAlign.LEFT)

    streaming_select = ft.Dropdown(
        options=[
            ft.dropdown.Option("AK", text="Amazon Kindle"),
            ft.dropdown.Option("F", text="Físico"),
        ],
        label="Selecione a streaming"
    )

    cadastrar_btn = ft.ElevatedButton("Cadastrar", on_click=cadastrar)

    page.views.append(
        ft.View(
            "/",
            controls=[
                nome_input,
                streaming_select,
                cadastrar_btn,
            ]
        )
    )
```

▼ Cadastrar

Crie a função cadastrar:

```
def cadastrar(e):
    data = {
        'nome': nome_input.value,
        'streaming': streaming_select.value,
        'categorias': []
    }

    response = requests.post('http://127.0.0.1:8000/api/livros/', json=data)
```

Desenvolva a função get_livros:

```
def get_livros():
    response = requests.get('http://127.0.0.1:8000/api/livros')
    return response.json()
```

Exiba os livros cadastrados:

```
lista_livros = ft.ListView()

def carregar_livros():
    lista_livros.controls.clear()
    for i in get_livros():
        lista_livros.controls.append(
            ft.Container(
                ft.Text(i['nome']),
                bgcolor=ft.colors.BLACK12,
                padding=15,
```

```

        alignment=ft.alignment.center,
        margin=3,
        border_radius=10,

    )
)
page.update()

carregar_livros()

page.views.append(
    ft.View(
        "/",
        controls=[
            nome_input,
            streaming_select,
            cadastrar_btn,
            lista_livros
        ]
    )
)

```

Em cada livro redirecione para a página de review:

```
on_click=lambda e: page.go(f"/review?id={i['id']}")
```

Altere o route_change:

```

from urllib.parse import urlparse, parse_qs

def route_change(e):
    page.views.clear()
    if page.route == "/":
        home_page()
    elif page.route.startswith("/review"):
        parsed_url = urlparse(page.route)
        query_params = parse_qs(parsed_url.query)
        livro_id = query_params.get('id', [None])[0]
        review_page(livro_id)

    page.update()

```

▼ Review

Crie a página de review_page:

```

def review_page(livro_id):
    nota_input = ft.TextField(label="Nota (inteiro)", text_align=ft.TextAlign.LEFT, value="0", width=100)
    comentario_input = ft.TextField(label="Comentário", multiline=True, expand=True)

    avaliar_btn = ft.ElevatedButton("Avaliar", on_click=avaliar)
    voltar_btn = ft.ElevatedButton("Voltar", on_click=lambda _: page.go('/'))
    page.views.append(
        ft.View(
            "/review",
            controls=[
                ft.Text("Review Page"),
                ft.Text(f"Detalhes do livro com ID: {livro_id}"),
                nota_input,
                comentario_input,
                avaliar_btn,
                voltar_btn
            ]
        )
    )
)

```

Crie a função avaliar:

```

def avaliar(e):
    data = {
        'nota': int(nota_input.value),
        'comentarios': comentario_input.value
    }

    try:
        response = requests.put(f'http://127.0.0.1:8000/api/livros/{livro_id}', json=data)
        if response.status_code == 200:
            page.snack_bar = ft.SnackBar(ft.Text("Avaliação enviada com sucesso!"))
        else:
            page.snack_bar = ft.SnackBar(ft.Text("Erro ao enviar a avaliação."))
        page.snack_bar.open = True
    except Exception as ex:
        page.snack_bar = ft.SnackBar(ft.Text(f"Erro de conexão: {ex}"))
        page.snack_bar.open = True

    page.update()

```