

# MOBILE APP | Projeto 4

Acesse diretamente pelo Notion:

```
https://grizzly-amaranthus-f6a.notion.site/MOBILE-APP-Projeto-4-1946cf8ea89f805b82b2fd6236212ad3?pvs=4
```

Primeiro passo vamos instalar as dependências:

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

Agora crie a mínima estrutura de um APP FLET:

```
import flet as ft
import requests

# URL base da API – ajuste conforme necessário
API_BASE_URL = "http://localhost:8000/api/treino"

def main(page: ft.Page):
    page.title = "Exemplo"
    page.vertical_alignment = ft.MainAxisAlignment.CENTER

    page.add()

ft.app(main)

if __name__ == "__main__":
    ft.app(target=main)
```

Crie os elementos para criar o aluno:

```
nome_field = ft.TextField(label="Nome")
email_field = ft.TextField(label="Email")
faixa_field = ft.TextField(label="Faixa")
data_nascimento_field = ft.TextField(label="Data de Nascimento (YYYY-MM-DD)")
create_result = ft.Text()

create_button = ft.ElevatedButton(text="Criar Aluno", on_click=criar_aluno_click)
```

Adicione a função para se conectar ao back-end:

```
def criar_aluno_click(e):
    payload = {
        "nome": nome_field.value,
        "email": email_field.value,
        "faixa": faixa_field.value,
        "data_nascimento": data_nascimento_field.value,
    }
    try:
        response = requests.post(API_BASE_URL + "/", json=payload)
        if response.status_code == 200:
            aluno = response.json()
            create_result.value = f"Aluno criado: {aluno}"
        else:
            create_result.value = f"Erro: {response.text}"
    except Exception as ex:
        create_result.value = f"Exceção: {ex}"
    page.update()
```

Adicione todos os elementos em uma coluna:

```
criar_aluno_tab = ft.Column(
    [
        nome_field,
        email_field,
        faixa_field,
        data_nascimento_field,
        create_button,
        create_result,
    ],
    scroll=True,
)
```

Agora cria abas de navegação:

```
tabs = ft.Tabs(
    selected_index=0,
    tabs=[
        ft.Tab(text="Criar Aluno", content=criar_aluno_tab),
    ]
)
```

E adicione as abas na página principal:

```
page.add(
    tabs
)
```

Agora vamos listar todos os alunos:

```
students_table = ft.DataTable(
    columns=[
        ft.DataColumn(ft.Text("ID")),
        ft.DataColumn(ft.Text("Nome")),
        ft.DataColumn(ft.Text("Email")),
        ft.DataColumn(ft.Text("Faixa")),
        ft.DataColumn(ft.Text("Data Nascimento")),
    ],
    rows=[],
)
list_result = ft.Text()

def listar_alunos_click(e):
    try:
        response = requests.get(API_BASE_URL + "/alunos/")
        if response.status_code == 200:
            alunos = response.json()
            # Limpa as linhas anteriores
            students_table.rows.clear()
            for aluno in alunos:
                row = ft.DataRow(
                    cells=[
                        ft.DataCell(ft.Text(str(aluno.get("id", "")))),
                        ft.DataCell(ft.Text(aluno.get("nome", ""))),
                        ft.DataCell(ft.Text(aluno.get("email", ""))),
                        ft.DataCell(ft.Text(aluno.get("faixa", ""))),
                        ft.DataCell(ft.Text(aluno.get("data_nascimento", ""))),
                    ]
                )
                students_table.rows.append(row)
            list_result.value = f"{len(alunos)} alunos encontrados."
        else:
            list_result.value = f"Erro: {response.text}"
    except Exception as ex:
        list_result.value = f"Exceção: {ex}"
    page.update()

list_button = ft.ElevatedButton(text="Listar Alunos", on_click=listar_alunos_click)
listar_alunos_tab = ft.Column([list_button, students_table, list_result], scroll=True)
```

Agora adicione dentro de tabs:

```
ft.Tab(text="Listar Alunos", content=listar_alunos_tab),
```

Para realizar a aula de algum aluno vamos desenvolver:

```
email_aula_field = ft.TextField(label="Email do Aluno")
qtd_field = ft.TextField(label="Quantidade de Aulas", value="1")
aula_result = ft.Text()

def marcar_aula_click(e):
    try:
        qtd = int(qtd_field.value)
        payload = {
            "qtd": qtd,
            "email_aluno": email_aula_field.value,
        }
        response = requests.post(API_BASE_URL + "/aula_realizada/", json=payload)
        if response.status_code == 200:
            # A API retorna uma mensagem de sucesso
            mensagem = response.json() # pode ser uma string ou objeto
            aula_result.value = f"Sucesso: {mensagem}"
        else:
            aula_result.value = f"Erro: {response.text}"
    except Exception as ex:
        aula_result.value = f"Exceção: {ex}"
    page.update()

aula_button = ft.ElevatedButton(text="Marcar Aula Realizada", on_click=marcar_aula_click)
aula_tab = ft.Column([email_aula_field, qtd_field, aula_button, aula_result], scroll=True)
```

Lembrando de adicionar em tabs!

```
ft.Tab(text="Aula realizada", content=aula_tab),
```

Para o progresso do aluno:

```
email_progress_field = ft.TextField(label="Email do Aluno")
progress_result = ft.Text()

def consultar_progresso_click(e):
    try:
```

```

email = email_progress_field.value
response = requests.get(
    API_BASE_URL + "/student_progress/", params={"student_email": email}
)
if response.status_code == 200:
    progress = response.json()
    progress_result.value = (
        f"Nome: {progress.get('student_name', '')}\n"
        f"Email: {progress.get('student_email', '')}\n"
        f"Faixa Atual: {progress.get('current_belt', '')}\n"
        f"Aulas Totais: {progress.get('total_lessons', 0)}\n"
        f"Aulas necessárias para a próxima faixa: {progress.get('lessons_needed_for_next_belt', 0)}"
    )
else:
    progress_result.value = f"Erro: {response.text}"
except Exception as ex:
    progress_result.value = f"Exceção: {ex}"
page.update()

progress_button = ft.ElevatedButton(text="Consultar Progresso", on_click=consultar_progresso_click)
progresso_tab = ft.Column([email_progress_field, progress_button, progress_result], scroll=True)

```

E para finalizar, vamos desenvolver a aba de atualizar aluno:

```

id_aluno_field = ft.TextField(label="ID do Aluno")
nome_update_field = ft.TextField(label="Novo Nome")
email_update_field = ft.TextField(label="Novo Email")
faixa_update_field = ft.TextField(label="Nova Faixa")
data_nascimento_update_field = ft.TextField(label="Nova Data de Nascimento (YYYY-MM-DD)")
update_result = ft.Text()

def atualizar_aluno_click(e):
    try:
        aluno_id = id_aluno_field.value
        if not aluno_id:
            update_result.value = "ID do aluno é necessário."
        else:
            payload = {}
            if nome_update_field.value:
                payload["nome"] = nome_update_field.value
            if email_update_field.value:
                payload["email"] = email_update_field.value
            if faixa_update_field.value:
                payload["faixa"] = faixa_update_field.value
            if data_nascimento_update_field.value:
                payload["data_nascimento"] = data_nascimento_update_field.value

            response = requests.put(API_BASE_URL + f"/alunos/{aluno_id}", json=payload)
            if response.status_code == 200:
                aluno = response.json()
                update_result.value = f"Aluno atualizado: {aluno}"
            else:
                update_result.value = f"Erro: {response.text}"
    except Exception as ex:
        update_result.value = f"Exceção: {ex}"
    page.update()

update_button = ft.ElevatedButton(text="Atualizar Aluno", on_click=atualizar_aluno_click)
atualizar_tab = ft.Column(
    [
        id_aluno_field,
        nome_update_field,
        email_update_field,
        faixa_update_field,
        data_nascimento_update_field,
        update_button,
        update_result,
    ],
    scroll=True,
)

```