

♦ 1. EXEMPLO DE HOSPITAL

Vamos cadastrar **pacientes** com nome, idade e doença.

♦ VERSÃO 1: Salvando com JSON

Arquivo: `pacientes.json`

json

CopiarEditar

```
[
  {
    "nome": "Ana",
    "idade": 30,
    "doenca": "Gripe"
  }
]
```

app.py com Flask + JSON:

python

CopiarEditar

```
from flask import Flask, request, jsonify
import json
import os

app = Flask(__name__)
ARQUIVO = 'pacientes.json'

def carregar_pacientes():
    if os.path.exists(ARQUIVO):
        with open(ARQUIVO, 'r') as f:
            return json.load(f)
    return []

def salvar_pacientes(pacientes):
    with open(ARQUIVO, 'w') as f:
        json.dump(pacientes, f, indent=4)

@app.route('/cadastrar', methods=['POST'])
def cadastrar():
```

```
    dados = request.json
    pacientes = carregar_pacientes()
    pacientes.append(dados)
    salvar_pacientes(pacientes)
    return jsonify({'mensagem': 'Paciente cadastrado!'})

@app.route('/listar')
def listar():
    pacientes = carregar_pacientes()
    return jsonify(pacientes)

if __name__ == '__main__':
    app.run(debug=True)
```

♦ **VERSÃO 2: Somente com lista e dicionário (temporário, sem salvar)**

```
python
CopiarEditar
from flask import Flask, request, jsonify

app = Flask(__name__)
pacientes = [] # Lista em memória

@app.route('/cadastrar', methods=['POST'])
def cadastrar():
    dados = request.json
    pacientes.append(dados)
    return jsonify({'mensagem': 'Paciente cadastrado!'})

@app.route('/listar')
def listar():
    return jsonify(pacientes)

if __name__ == '__main__':
    app.run(debug=True)
```

♦ **2. EXEMPLO DE BIBLIOTECA**

Vamos cadastrar **livros** com título, autor e ano.

♦ VERSÃO 1: Salvando em JSON

Arquivo: `livros.json`

json

CopiarEditar

```
[
  {
    "titulo": "Dom Casmurro",
    "autor": "Machado de Assis",
    "ano": 1899
  }
]
```

app.py com Flask + JSON:

python

CopiarEditar

```
from flask import Flask, request, jsonify
import json
import os

app = Flask(__name__)
ARQUIVO = 'livros.json'

def carregar_livros():
    if os.path.exists(ARQUIVO):
        with open(ARQUIVO, 'r') as f:
            return json.load(f)
    return []

def salvar_livros(livros):
    with open(ARQUIVO, 'w') as f:
        json.dump(livros, f, indent=4)

@app.route('/cadastrar', methods=['POST'])
def cadastrar():
    dados = request.json
    livros = carregar_livros()
```

```
        livros.append(dados)
        salvar_livros(livros)
        return jsonify({'mensagem': 'Livro cadastrado!'})

@app.route('/listar')
def listar():
    livros = carregar_livros()
    return jsonify(livros)

if __name__ == '__main__':
    app.run(debug=True)
```

♦ VERSÃO 2: Somente com listas e dicionários

```
python
CopiarEditar
from flask import Flask, request, jsonify

app = Flask(__name__)
livros = [] # Lista em memória

@app.route('/cadastrar', methods=['POST'])
def cadastrar():
    dados = request.json
    livros.append(dados)
    return jsonify({'mensagem': 'Livro cadastrado!'})

@app.route('/listar')
def listar():
    return jsonify(livros)

if __name__ == '__main__':
    app.run(debug=True)
```

✓ RESUMO DAS DIFERENÇAS

Recurso	JSON	Somente em memória
---------	------	--------------------

Dados persistem após reiniciar	<input checked="" type="checkbox"/> Sim	<input type="checkbox"/> Não
--------------------------------	---	------------------------------

Fácil de implementar	Médio	<input checked="" type="checkbox"/> Simples
----------------------	-------	---

Boa para testes rápidos	<input type="checkbox"/>	<input checked="" type="checkbox"/>
-------------------------	--------------------------	-------------------------------------

Boa para produção simples	<input checked="" type="checkbox"/>	<input type="checkbox"/>
---------------------------	-------------------------------------	--------------------------