

## Vamos comparar as abordagens:

Forma de salvar	Onde ficam os dados	Persistem se reiniciar o servidor?
✅ Listas e dicionários (em memória)	Apenas enquanto o app Flask estiver rodando	❌ Não persistem
✅ Arquivos JSON ou banco de dados	Salvos no disco (HD/SSD)	✅ Sim, persistem

---

## Quando usar listas e dicionários?

- Durante o **desenvolvimento ou testes** rápidos.
  - Se você **não precisa salvar os dados permanentemente**.
  - Para **provas ou exercícios** em que o foco é entender o funcionamento e não a persistência.
- 

## ✅ Exemplo simples usando só dicionários:

python

CopiarEditar

```
from flask import Flask, request, session, redirect, url_for
```

```
app = Flask(__name__)
```

```
app.secret_key = 'segredo123'
```

```
usuarios = {  
    'alice': {'senha': '123', 'carrinho': []},  
    'bob': {'senha': '456', 'carrinho': []}  
}
```

```
@app.route('/login', methods=['POST'])
```

```
def login():
```

```
    nome = request.form['nome']
```

```
    senha = request.form['senha']
```

```
    if nome in usuarios and usuarios[nome]['senha'] == senha:
```

```
        session['usuario'] = nome
```

```
        return redirect(url_for('mostrar_carrinho'))
```

```
        return 'Login inválido'

@app.route('/adicionar/<produto>')
def adicionar(produto):
    usuario = session.get('usuario')
    if usuario:
        usuarios[usuario]['carrinho'].append(produto)
        return f'{produto} adicionado ao carrinho!'
    return 'Faça login primeiro'

@app.route('/carrinho')
def mostrar_carrinho():
    usuario = session.get('usuario')
    if usuario:
        return f"Carrinho de {usuario}:  
{usuarios[usuario]['carrinho']}"
    return 'Faça login primeiro'
```

---

### ⚠ Cuidados

- Se você reiniciar o Flask (**Ctrl+C** e **flask run** de novo), **os dicionários e listas voltam a estar vazios**.
  - Isso significa que **nada será salvo de verdade**. Serve apenas enquanto o app estiver rodando.
- 

### ✅ Então, posso usar só listas e dicionários?

**Pode sim!** Só precisa lembrar que **isso é temporário**, e que para **sistemas reais** ou que **exigem guardar dados entre acessos**, você precisaria de algo como:

- **JSON** (mais simples),
- ou um **banco de dados** (**SQLite**, **PostgreSQL**, etc).

Se quiser, posso montar um exemplo completo (tipo lojinha com carrinho) **só com listas/dicionários**, ou comparar com uma versão que salva em JSON.

Quer que eu faça isso?

Perguntar ao ChatGPT

**Você disse:**

quero

**O ChatGPT disse:**

Perfeito! Vamos fazer **um sisteminha de loja com carrinho**, com **login, produtos e compras, sem usar JSON**, apenas com **listas e dicionários na memória**.

---

### ✅ Funcionalidades:

- Tela de login (usuários pré-cadastrados)
  - Listagem de produtos
  - Adição ao carrinho
  - Visualização do carrinho
  - Finalizar compra (esvazia o carrinho)
- 

### Código completo (Flask — tudo em memória)

python

CopiarEditar

```
from flask import Flask, render_template_string, request, redirect, session, url_for
```

```
app = Flask(__name__)
app.secret_key = 'chave_secreta'
```

```
# Usuários e carrinhos em memória
usuarios = {
    'ana': {'senha': '123', 'carrinho': []},
    'joao': {'senha': '456', 'carrinho': []}
}
```

```
# Produtos da loja
produtos = [
    {'id': 1, 'nome': 'Livro', 'preco': 30},
    {'id': 2, 'nome': 'Caneta', 'preco': 5},
```

```

        {'id': 3, 'nome': 'Caderno', 'preco': 20}
    ]

# Template base
base_html = '''
<!DOCTYPE html>
<html>
<head><title>{{ title }}</title></head>
<body>
    <h1>{{ title }}</h1>
    {% with mensagens = get_flashed_messages() %}
        {% if mensagens %}
            <ul>{% for msg in mensagens %}<li>{{ msg }}</li>{% endfor
%}</ul>
        {% endif %}
    {% endwith %}
    {% block conteudo %}{% endblock %}
</body>
</html>
'''

@app.route('/')
def home():
    return redirect(url_for('login'))

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        nome = request.form['usuario']
        senha = request.form['senha']
        if nome in usuarios and usuarios[nome]['senha'] == senha:
            session['usuario'] = nome
            return redirect(url_for('loja'))
        return 'Login inválido'
    return render_template_string(base_html + '''
    {% block conteudo %}
        <form method="POST">
            Usuário: <input name="usuario"><br>
            Senha: <input name="senha" type="password"><br>
            <button type="submit">Entrar</button>
        </form>
    ''')

```

```

        {% endblock %}
        '', title='Login')

@app.route('/loja')
def loja():
    if 'usuario' not in session:
        return redirect(url_for('login'))
    return render_template_string(base_html + '''
    {% block conteudo %}
    <p>Bem-vindo, {{ session['usuario'] }}!</p>
    <h2>Produtos disponíveis:</h2>
    <ul>
        {% for p in produtos %}
        <li>{{ p.nome }} - R$ {{ p.preco }}
            <a href="{{ url_for('adicionar', id=p.id)
}}">[Adicionar]</a>
        </li>
        {% endfor %}
    </ul>
    <a href="{{ url_for('ver_carrinho') }}">Ver carrinho</a> |
    <a href="{{ url_for('logout') }}">Sair</a>
    {% endblock %}
    '', title='Loja', produtos=produtos)

@app.route('/adicionar/<int:id>')
def adicionar(id):
    if 'usuario' not in session:
        return redirect(url_for('login'))
    usuario = session['usuario']
    for p in produtos:
        if p['id'] == id:
            usuarios[usuario]['carrinho'].append(p)
            break
    return redirect(url_for('ver_carrinho'))

@app.route('/carrinho')
def ver_carrinho():
    if 'usuario' not in session:
        return redirect(url_for('login'))
    usuario = session['usuario']
    carrinho = usuarios[usuario]['carrinho']

```

```

total = sum(p['preco'] for p in carrinho)
return render_template_string(base_html + '''
{% block conteudo %}
<h2>Carrinho de {{ session['usuario'] }}</h2>
{% if carrinho %}
    <ul>
        {% for item in carrinho %}
            <li>{{ item.nome }} - R$ {{ item.preco }}</li>
        {% endfor %}
    </ul>
    <p>Total: R$ {{ total }}</p>
    <a href="{{ url_for('finalizar') }}">Finalizar compra</a>
{% else %}
    <p>Seu carrinho está vazio.</p>
{% endif %}
<br><a href="{{ url_for('loja') }}">Voltar à loja</a>
{% endblock %}
''', title='Carrinho', carrinho=carrinho, total=total)

@app.route('/finalizar')
def finalizar():
    if 'usuario' not in session:
        return redirect(url_for('login'))
    usuario = session['usuario']
    usuarios[usuario]['carrinho'].clear()
    return render_template_string(base_html + '''
{% block conteudo %}
<p>Compra finalizada com sucesso!</p>
<a href="{{ url_for('loja') }}">Voltar à loja</a>
{% endblock %}
''', title='Compra Concluída')

@app.route('/logout')
def logout():
    session.pop('usuario', None)
    return redirect(url_for('login'))

```



**O que esse sistema faz:**

- Funciona só enquanto o Flask estiver rodando.
- Carrinho fica salvo apenas em dicionário (**usuarios**).
- Se reiniciar o servidor, tudo volta ao início.
- Fácil de entender, ótimo para **provas e aprendizado**.

---

Se quiser depois, posso mostrar como **transformar isso para salvar com JSON** ou com **banco de dados (tipo SQLite)**, passo a passo.

Quer ver isso também?

Ferramentas

O