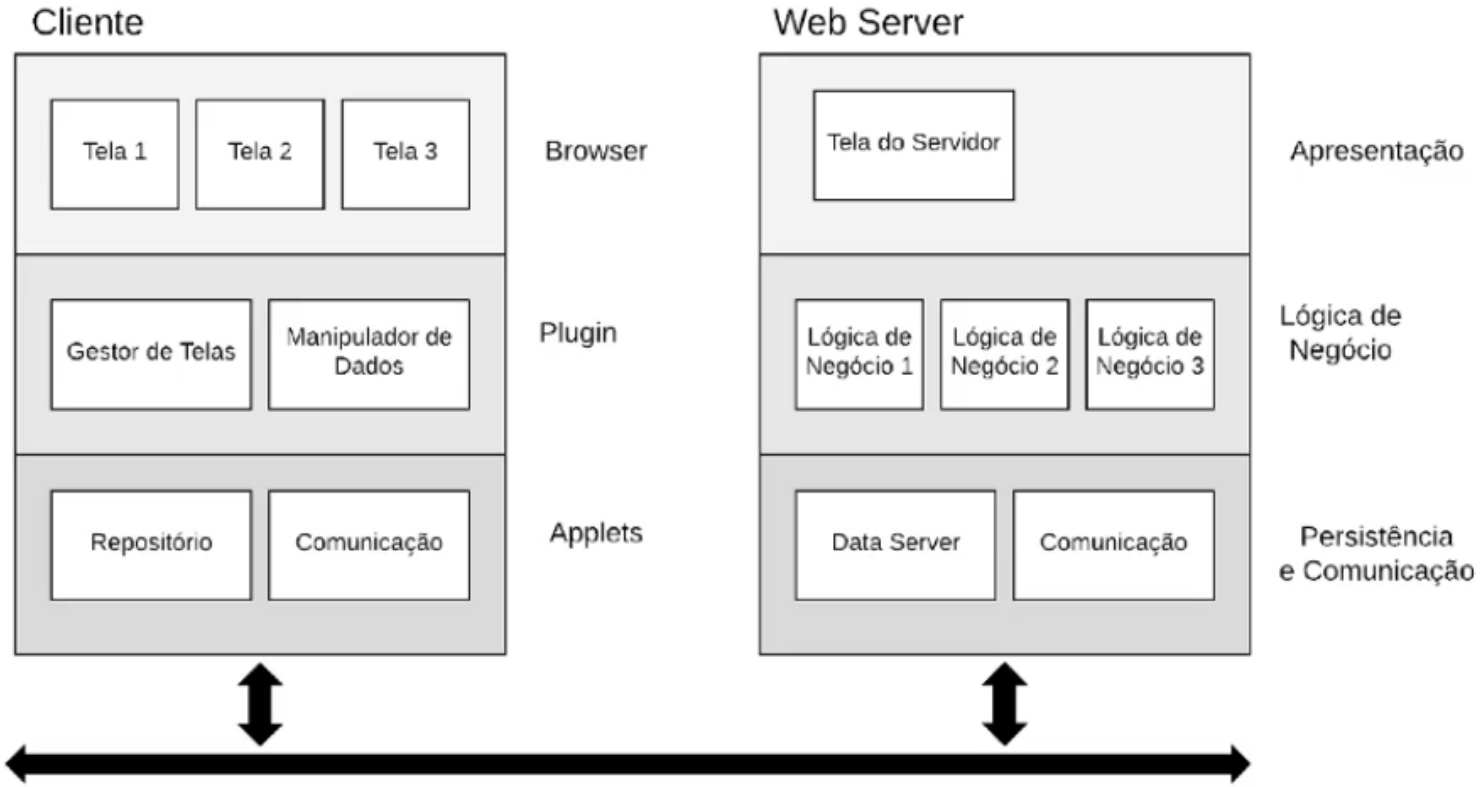
Arquitetura MVC

Michelet del Carpio chávez

Agenda

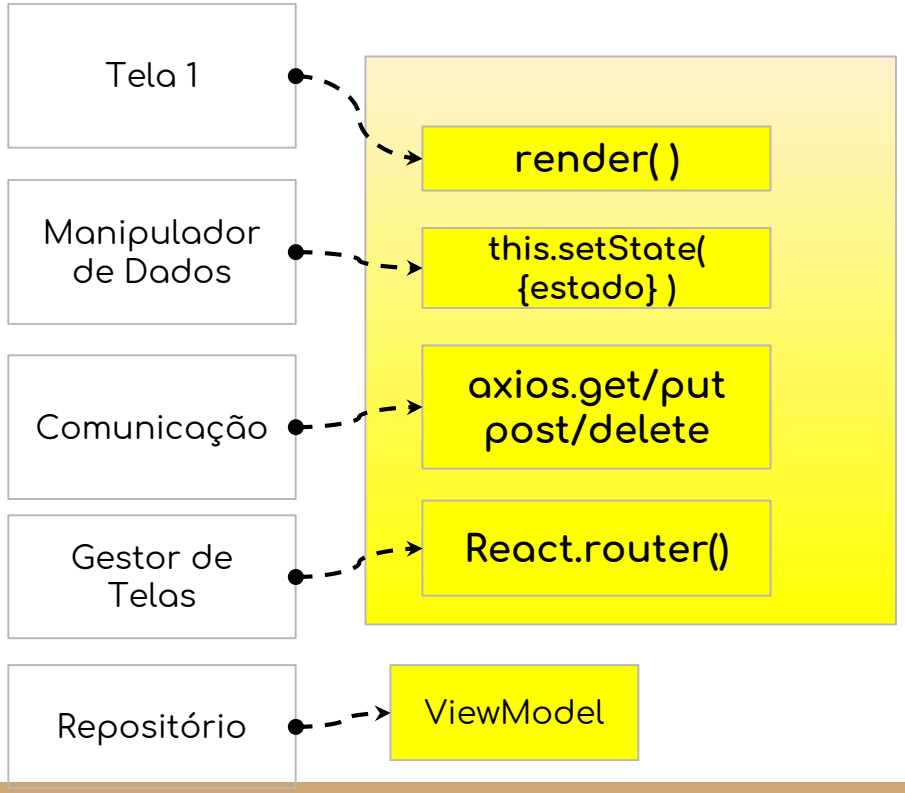
- Revisão da arquitetura da aplicação
  - Cliente
  - Servidor
- Projeto usando Flask
  - Controller
  - Model
  - View
- Mini-tutorial de Flask
  - flask
  - flask-rest-api
- Demo

Arquitetura da Aplicação

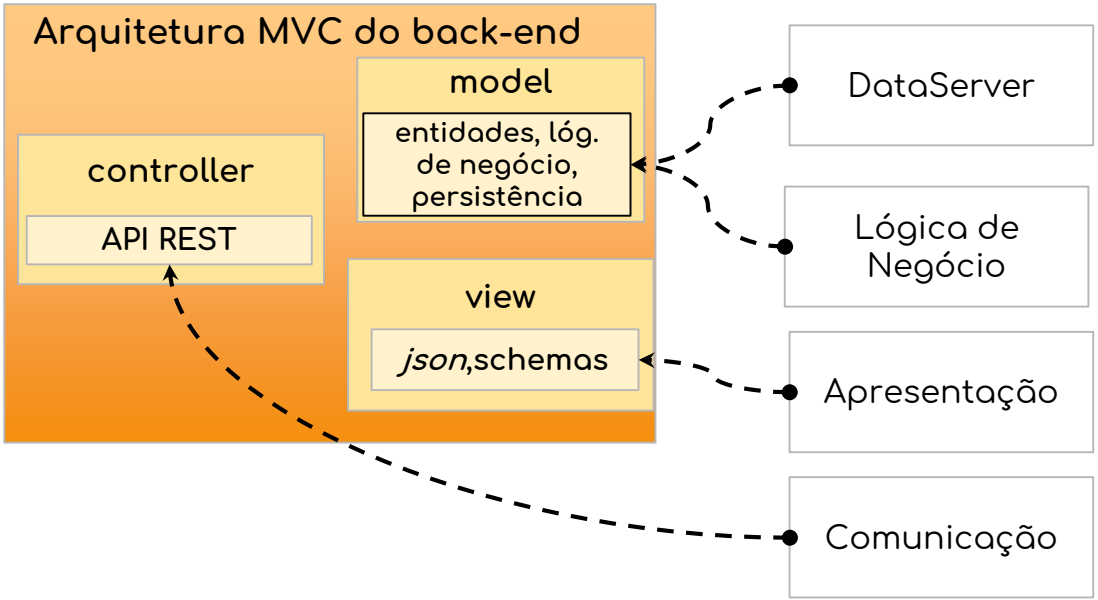


Mapeamento de responsabilidades do Cliente

Elementos da Arquitetura do Cliente

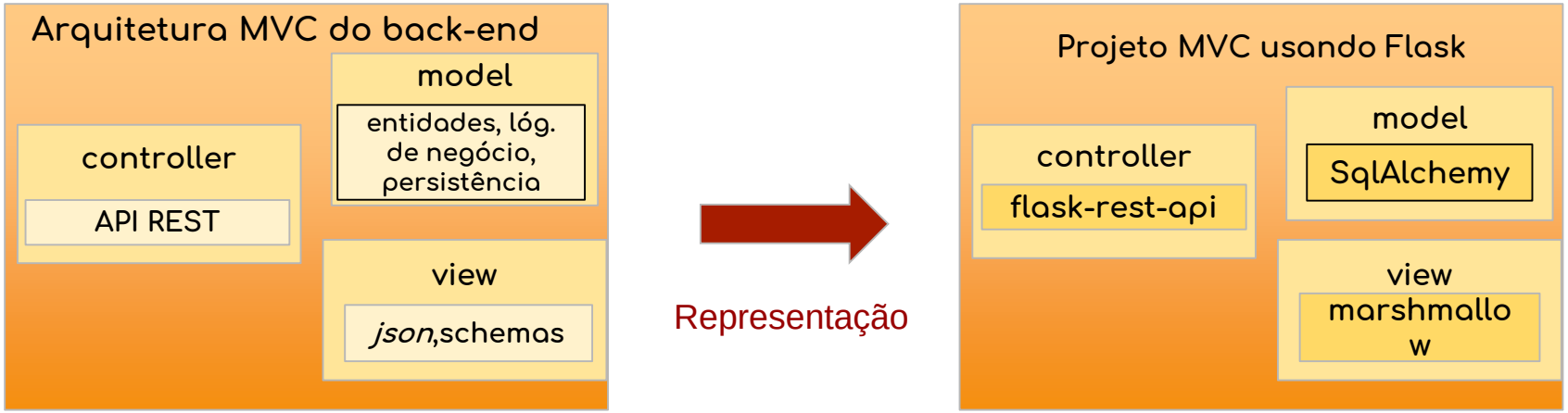


Mapeamento de responsabilidades do Servidor



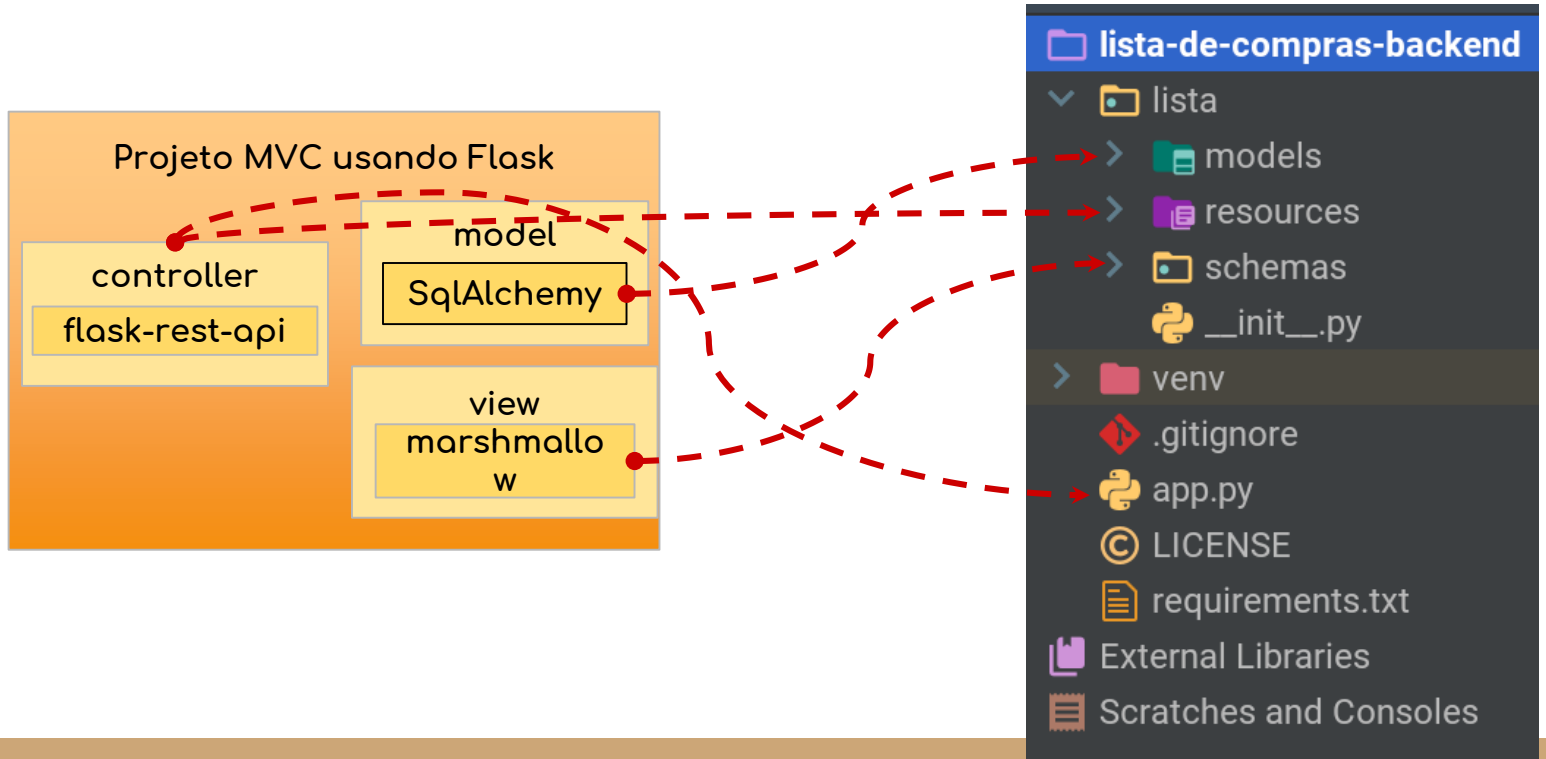
Elementos da Arquitetura do Servidor

Projeto da Arquitetura usando Flask

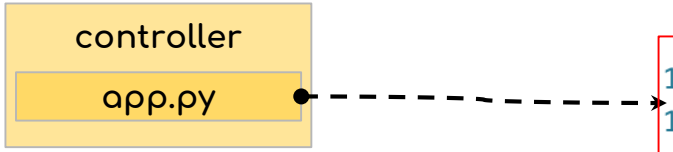


Mapeamento do Projeto MVC na Implementação

estrutura do código fonte

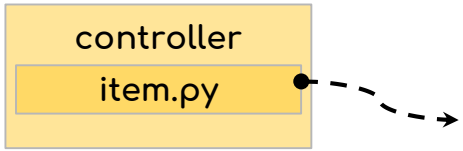


Projeto - Controller v1



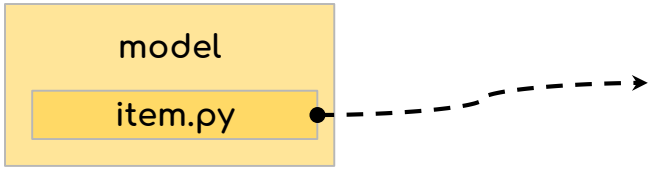

```
4   from flask import Flask
5   from flask_restful import Api
6   from lista.resources.api_v1 import Item,ItemList
7   app = Flask(__name__)
8   api = Api(app)
9
10  api.add_resource(ItemList, '/itens')
11  api.add_resource(Item, '/item','/item/<string:item>')
```

Projeto - Controller v1




```
Esta classe representa um recurso na arquitetura REST.
Ela implementa os métodos GET,POST,PUT e DELETE
'''
class Item(Resource):
    def get(self,item):
        if item not in LISTA_DE_COMPRAS:
            abort(404, message="Item {} não está na lista".format(item))
        return {"item":item},200
    def post(self):
        data = _item_parser.parse_args()
        item = data['item']
        if item in LISTA_DE_COMPRAS :
            return {"message":"Item {} já está na lista".format(item)}
        else:
            LISTA_DE_COMPRAS.append(item)
        return item, 201
```

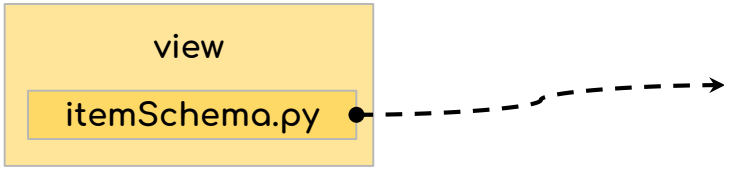
Projeto - Model v2



item.py

```
1   from datetime import datetime
2   class Item(object):
3       def __init__(self,id,nome):
4           self.id = id
5           self.nome = nome
6           self.dataCriacao = datetime.now()
```

Projeto - View v2



ItemSchema.py

```python
from marshmallow import Schema,fields
from lista.models.item import Item

class ItemSchema(Schema):
    nome = fields.Str()
    id = fields.Integer()
    dataCriacao = fields.DateTime()
    class Meta:
        model = Item
```

Mini-tutorial de Flask - I

Referência: https://code.visualstudio.com/docs/python/tutorial-flask

Instalar python3

Criar um ambiente de projeto para python

```
# macOS/Linux

 pip3 install virtualenv
#linux

sudo apt-get install python3-venv    # If needed
python3 -m venv env

# Windows
py -3 -m venv env
```

Mini-tutorial de Flask - II

Ativar o ambiente

`source env/bin/activate`

`env\scripts\activate` (Windows)

Instalar o flask

```
# macOS/Linux
pip3 install flask

# Windows
pip install flask
```

Mini-tutorial de Flask - III

Editar o arquivo app.py com o seguinte conteúdo

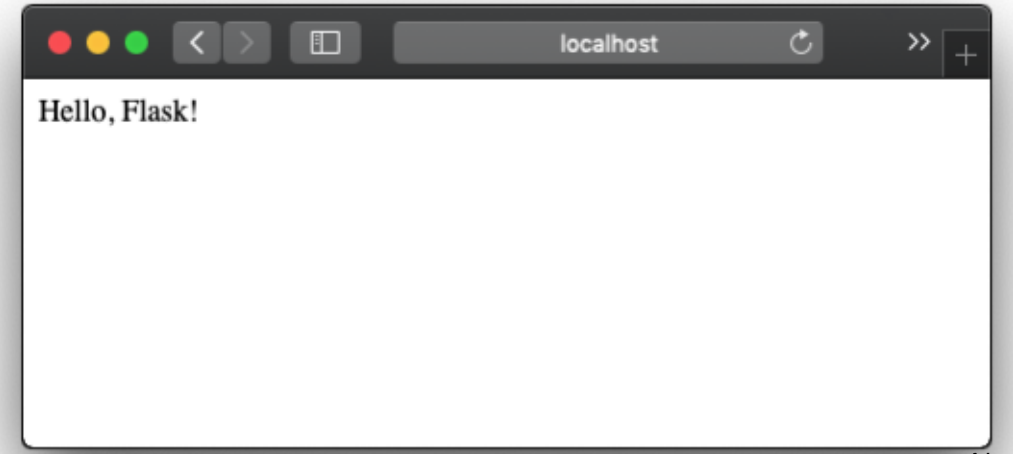
app.py

```
1  from flask import Flask
2  app = Flask(__name__)
3  @app.route("/")
4  def home():
5      return 'Hello, Flask!'
```

Executar o programa:

`python3 -m flask run` (MacOS/Linux)

`python -m flask run` (Windows)



Tutorial Flask Rest - API

Referência https://flask-restful.readthedocs.io/en/latest/quickstart.html

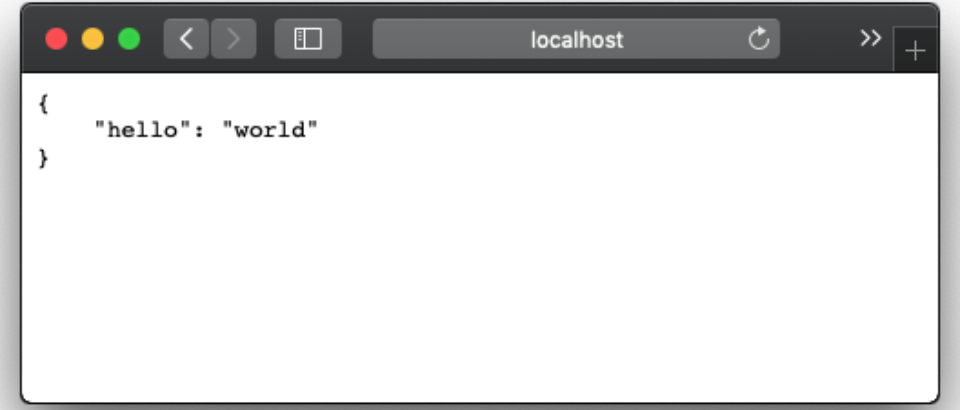
app.py

```python
from flask import Flask
from flask_restful import Resource, Api
app = Flask(__name__)
api = Api(app)

#controller, model & view
class HelloWorld(Resource):
    def get(self):
        return {'hello': 'world'}

api.add_resource(HelloWorld, '/')

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

No Terminal, executar:

`python3 app.py` (MacOS/Linux)

`python app.py` (Windows)



Referências

Repositório da Lista de Compras usando flask: https://github.com/miklt/lista-de-compras-backend

Flask: https://code.visualstudio.com/docs/python/tutorial-flask

Flask Rest API https://flask-restful.readthedocs.io/en/latest/quickstart.html

Demo