

## Listar Atividades

The image shows a development environment with VS Code on the left and Postman on the right. In VS Code, the `atividade_controller.py` file is open, showing the `listar_atividades` method in the `AtividadeController` class. The method calls `atividade_model.listar_atividades()` and returns the result as JSON. The terminal shows the application running on `http://localhost:5002`. In Postman, the `GET` request to `localhost:5002/atividades` is shown, returning a 200 OK status. The response body is a JSON array of activities.

```
1 from flask import Blueprint, jsonify, request
2 from models import atividade_model
3 from clients.pessoa_service_client import PessoaServiceClient
4
5 atividade_bp = Blueprint('atividade_bp', __name__)
6
7 @atividade_bp.route('/', methods=['GET'])
8 def listar_atividades():
9     atividades = atividade_model.listar_atividades()
10    return jsonify(atividades)
11
12 @atividade_bp.route('/<int:id_atividade>', methods=['GET'])
13 def obter_atividade(id_atividade):
14     try:
15         atividade = atividade_model.obter_atividade(id_atividade)
16         return jsonify(atividade)
17     except atividade_model.AtividadeNotFound:
18         return jsonify({'erro': 'Atividade não encontrada'}), 404
19
20 @atividade_bp.route('/', methods=['POST'])
21 def criar_atividade():
22     data = request.get_json()
23     id_disciplina = data.get('id_disciplina')
24     enunciado = data.get('enunciado')
25     respostas = data.get('respostas', [])
26
27     nova_atividade = atividade_model.criar_atividade(id_disciplina, enunciado, respostas)
28     return jsonify(nova_atividade), 201
29
30 @atividade_bp.route('/<int:id_atividade>', methods=['PUT'])
31 def atualizar_atividade(id_atividade):
32     data = request.get_json()
33     id_disciplina = data.get('id_disciplina')
```

Postman Request: `GET localhost:5002/atividades`

Response Body (JSON):

```
{
  "enunciado": "Crie um app de todo em Flask",
  "id_atividade": 1,
  "id_disciplina": 1,
  "respostas": [
    {
      "id_aluno": 1,
      "nota": 9,
      "resposta": "todo.py"
    },
    {
      "id_aluno": 2,
      "resposta": "todo.zip.rar"
    },
    {
      "id_aluno": 4,
      "nota": 10,
      "resposta": "todo.zip"
    }
  ]
}
```

## Obter Atividade por ID

The image shows a development environment with VS Code on the left and Postman on the right. In VS Code, the `atividade_controller.py` file is open, showing the `obter_atividade` method in the `AtividadeController` class. The method calls `atividade_model.obter_atividade(id_atividade)` and returns the result as JSON. The terminal shows the application running on `http://localhost:5002`. In Postman, the `GET` request to `localhost:5002/atividades/1` is shown, returning a 200 OK status. The response body is a JSON object representing a single activity.

```
1 from flask import Blueprint, jsonify, request
2 from models import atividade_model
3 from clients.pessoa_service_client import PessoaServiceClient
4
5 atividade_bp = Blueprint('atividade_bp', __name__)
6
7 @atividade_bp.route('/', methods=['GET'])
8 def listar_atividades():
9     atividades = atividade_model.listar_atividades()
10    return jsonify(atividades)
11
12 @atividade_bp.route('/<int:id_atividade>', methods=['GET'])
13 def obter_atividade(id_atividade):
14     try:
15         atividade = atividade_model.obter_atividade(id_atividade)
16         return jsonify(atividade)
17     except atividade_model.AtividadeNotFound:
18         return jsonify({'erro': 'Atividade não encontrada'}), 404
19
20 @atividade_bp.route('/', methods=['POST'])
21 def criar_atividade():
22     data = request.get_json()
23     id_disciplina = data.get('id_disciplina')
24     enunciado = data.get('enunciado')
25     respostas = data.get('respostas', [])
26
27     nova_atividade = atividade_model.criar_atividade(id_disciplina, enunciado, respostas)
28     return jsonify(nova_atividade), 201
29
30 @atividade_bp.route('/<int:id_atividade>', methods=['PUT'])
31 def atualizar_atividade(id_atividade):
32     data = request.get_json()
33     id_disciplina = data.get('id_disciplina')
```

Postman Request: `GET localhost:5002/atividades/1`

Response Body (JSON):

```
{
  "enunciado": "Crie um app de todo em Flask",
  "id_atividade": 1,
  "id_disciplina": 1,
  "respostas": [
    {
      "id_aluno": 1,
      "nota": 9,
      "resposta": "todo.py"
    },
    {
      "id_aluno": 2,
      "resposta": "todo.zip.rar"
    },
    {
      "id_aluno": 4,
      "nota": 10,
      "resposta": "todo.zip"
    }
  ]
}
```

## Criar Atividade

The image displays two screenshots of a development environment, likely Visual Studio Code, showing the implementation of a REST API for creating and listing activities.

**Top Screenshot:**

- Left Panel (Code Editor):** Shows the `atividade_controller.py` file. The `criar_atividade` method is implemented, which takes a request, extracts the discipline, enunciado, and respostas, and then calls `atividade_model.criar_atividade` to create a new activity. The response is a 201 status code.
- Right Panel (API Network):** Shows the `Atividade Service Testes / Criar Atividade` endpoint. The request is a POST to `localhost:5002/atividades` with a JSON body: 

```
{ "id_disciplina": 1, "enunciado": "Nova atividade de teste", "respostas": [] }
```

. The response is a 201 status code, indicating the activity was created successfully.

**Bottom Screenshot:**

- Left Panel (Code Editor):** Shows the `atividade_controller.py` file. The `listar_atividades` method is implemented, which calls `atividade_model.listar_atividades` to retrieve all activities. The response is a 200 status code.
- Right Panel (API Network):** Shows the `Atividade Service Testes / Listar Atividades` endpoint. The request is a GET to `localhost:5002/atividades`. The response is a 200 status code, indicating the activities were retrieved successfully. The response body shows a list of activities, including the one created in the previous screenshot.

## Atualizar Atividade

The image displays a development environment with two main windows. The left window shows the source code for a Flask application, and the right window shows the API Network client interface.

**Left Window (Code Editor):**

```
1 from flask import Blueprint, jsonify, request
2 from models import atividade_model
3 from clients.pessoa_service_client import PessoaServiceClient
4
5 atividade_bp = Blueprint('atividade_bp', __name__)
6
7 @atividade_bp.route('/', methods=['GET'])
8 def listar_atividades():
9     atividades = atividade_model.listar_atividades()
10    return jsonify(atividades)
11
12 @atividade_bp.route('/<int: id_atividade>', methods=['GET'])
13 def obter_atividade(id_atividade):
14     try:
15         atividade = atividade_model.obter_atividade(id_atividade)
16         return jsonify(atividade)
17     except atividade_model.AtividadeNotFound:
18         return jsonify({'erro': 'Atividade não encontrada'}), 404
19
20 @atividade_bp.route('/', methods=['POST'])
21 def criar_atividade():
22     data = request.get_json()
23     id_disciplina = data.get('id_disciplina')
24     enunciado = data.get('enunciado')
25     respostas = data.get('respostas', [])
26
27     nova_atividade = atividade_model.criar_atividade(id_disciplina, enunciado, respostas)
28     return jsonify(nova_atividade), 201
29
30 @atividade_bp.route('/<int: id_atividade>', methods=['PUT'])
31 def atualizar_atividade(id_atividade):
32     data = request.get_json()
33     id_disciplina = data.get('id_disciplina')
34     enunciado = data.get('enunciado')
```

**Right Window (API Network):**

The API Network client shows a PUT request to `localhost:5002/atividades/1` with a 200 OK status. The response body is:

```
1 {
2   "id_disciplina": 1, "enunciado": "Atividade de teste atualizada",
3   "id_atividade": 1,
4   "id_disciplina": 1,
5   "respostas": [
6     {
7       "id_aluno": 1,
8       "nota": 8,
9       "resposta": "resposta1"
10    }
11  ]
12 }
```

The bottom window shows the same code editor and API Network client, but with a GET request to `localhost:5002/atividades` and a 200 OK status. The response body is:

```
1 [
2   {
3     "enunciado": "Atividade de teste atualizada",
4     "id_atividade": 1,
5     "id_disciplina": 1,
6     "respostas": [
7       {
8         "id_aluno": 1,
9         "nota": 8,
10        "resposta": "resposta1"
11      }
12    ]
13  },
14  {
15    "enunciado": "Crie um servidor que envia email em Flask",
16    "id_atividade": 2,
17    "id_disciplina": 1,
18    "respostas": [
19      {
20        "id_aluno": 4,
21        "nota": 10,
22        "resposta": "email.zip"
23      }
24    ]
25  },
26  {
27    "enunciado": "Nova atividade de teste",
28    "id_atividade": 3,
29    "id_disciplina": 1,
30    "respostas": []
31  }
32 ]
```

# Excluir Atividade

File Edit Selection

Flask-microservice-mvc

personaservice\_client.py atividade\_controller.py config.py atividade\_model.py M

atividade\_controller.py

1 from flask import Blueprint, jsonify, request  
2 from models import atividade\_model  
3 from clients.pessoa\_service\_client import PessoaServiceClient  
4  
5 atividade\_bp = Blueprint('atividade\_bp', \_\_name\_\_)  
6  
7 @atividade\_bp.route('/', methods=['GET'])  
8 def listar\_atividades():  
9 atividades = atividade\_model.listar\_atividades()  
10 return jsonify(atividades)  
11  
12 @atividade\_bp.route('/<int:id\_atividade>', methods=['GET'])  
13 def obter\_atividade(id\_atividade):  
14 try:  
15 atividade = atividade\_model.obter\_atividade(id\_atividade)  
16 return jsonify(atividade)  
17 except atividade\_model.AtividadeNotFound:  
18 return jsonify({'erro': 'Atividade não encontrada'}), 404  
19  
20 @atividade\_bp.route('/', methods=['POST'])  
21 def criar\_atividade():  
22 data = request.get\_json()  
23 id\_disciplina = data.get('id\_disciplina')  
24 enunciado = data.get('enunciado')  
25 respostas = data.get('respostas', [])  
26  
27 nova\_atividade = atividade\_model.criar\_atividade(id\_disciplina, enunciado, respostas)  
28 return jsonify(nova\_atividade), 201  
29  
30 @atividade\_bp.route('/<int:id\_atividade>', methods=['PUT'])  
31 def atualizar\_atividade(id\_atividade):  
32 data = request.get\_json()  
33 id\_disciplina = data.get('id\_disciplina')  
34 enunciado = data.get('enunciado')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

python3.12 - atividade\_service

127.0.0.1 - - [22/Nov/2024 09:56:11] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:56:11] "GET /atividades/1 HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:57:53] "GET /atividades/1 HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 09:58:35] "POST /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:58:35] "POST /atividades HTTP/1.1" 201 -  
127.0.0.1 - - [22/Nov/2024 09:58:53] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:58:53] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:58:53] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:59:33] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:59:33] "GET /atividades HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 09:59:52] "PUT /atividades/1 HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 09:59:52] "PUT /atividades/1 HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 10:00:01] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 10:00:01] "GET /atividades HTTP/1.1" 200 -

Personal Workspace

API Network

Atividade Service Testes / Excluir Atividade

DELETE localhost:5002/atividades/1

Send

Body

none

This request does not have a body

204 NO CONTENT 3 ms 153 B

Pretty Raw Preview Visualize JSON

1

File Edit Selection

Flask-microservice-mvc

personaservice\_client.py atividade\_controller.py config.py atividade\_model.py M

atividade\_controller.py

1 from flask import Blueprint, jsonify, request  
2 from models import atividade\_model  
3 from clients.pessoa\_service\_client import PessoaServiceClient  
4  
5 atividade\_bp = Blueprint('atividade\_bp', \_\_name\_\_)  
6  
7 @atividade\_bp.route('/', methods=['GET'])  
8 def listar\_atividades():  
9 atividades = atividade\_model.listar\_atividades()  
10 return jsonify(atividades)  
11  
12 @atividade\_bp.route('/<int:id\_atividade>', methods=['GET'])  
13 def obter\_atividade(id\_atividade):  
14 try:  
15 atividade = atividade\_model.obter\_atividade(id\_atividade)  
16 return jsonify(atividade)  
17 except atividade\_model.AtividadeNotFound:  
18 return jsonify({'erro': 'Atividade não encontrada'}), 404  
19  
20 @atividade\_bp.route('/', methods=['POST'])  
21 def criar\_atividade():  
22 data = request.get\_json()  
23 id\_disciplina = data.get('id\_disciplina')  
24 enunciado = data.get('enunciado')  
25 respostas = data.get('respostas', [])  
26  
27 nova\_atividade = atividade\_model.criar\_atividade(id\_disciplina, enunciado, respostas)  
28 return jsonify(nova\_atividade), 201  
29  
30 @atividade\_bp.route('/<int:id\_atividade>', methods=['PUT'])  
31 def atualizar\_atividade(id\_atividade):  
32 data = request.get\_json()  
33 id\_disciplina = data.get('id\_disciplina')  
34 enunciado = data.get('enunciado')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

python3.12 - atividade\_service

127.0.0.1 - - [22/Nov/2024 09:58:35] "POST /atividades HTTP/1.1" 201 -  
127.0.0.1 - - [22/Nov/2024 09:58:53] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:58:53] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:59:33] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:59:33] "GET /atividades HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 09:59:52] "PUT /atividades/1 HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 09:59:52] "PUT /atividades/1 HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 10:00:01] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 10:00:01] "GET /atividades HTTP/1.1" 200 -  
127.0.0.1 - - [22/Nov/2024 10:00:34] "DELETE /atividades/1 HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 10:00:34] "DELETE /atividades/1 HTTP/1.1" 204 -  
127.0.0.1 - - [22/Nov/2024 10:00:42] "GET /atividades HTTP/1.1" 308 -  
127.0.0.1 - - [22/Nov/2024 10:00:42] "GET /atividades HTTP/1.1" 200 -

Personal Workspace

API Network

Atividade Service Testes / Listar Atividades

GET localhost:5002/atividades

Send

Params

Query Params

Key Value Des... Bulk Edit

Body

200 OK 2 ms 523 B

Pretty Raw Preview Visualize JSON

1 {  
2 {  
3 "enunciado": "Crie um servidor que envia email em  
4 Flask",  
5 "id\_atividade": 2,  
6 "id\_disciplina": 1,  
7 "respostas": [  
8 {  
9 "id\_aluno": 4,  
10 "nota": 10,  
11 "resposta": "email.zip"  
12 }  
13 ],  
14 }  
15 {  
16 "enunciado": "Nova atividade de teste",  
17 "id\_atividade": 3,  
18 "id\_disciplina": 1,  
19 "respostas": []  
20 }  
21 }

## Obter Atividade para Professor

The image shows a development environment with two main windows. The left window displays the source code for a Flask API, and the right window shows the Postman interface for testing the API.

**Source Code (Left Window):**

```
11 @atividade_bp.route('/<int:id_atividade>', methods=['GET'])
12 def obter_atividade(id_atividade):
13     try:
14         atividade = atividade_model.obter_atividade(id_atividade)
15         return jsonify(atividade)
16     except atividade_model.AtividadeNotFound:
17         return jsonify({'erro': 'Atividade não encontrada'}), 404
18
19 @atividade_bp.route('/<int:id_atividade>/professor/<int:id_professor>', methods=['GET'])
20 def obter_atividade_para_professor(id_atividade, id_professor):
21     try:
22         atividade = atividade_model.obter_atividade(id_atividade)
23         if not PessoaServiceClient.verificar_leciona(id_professor, atividade['id_disciplina']):
24             atividade = atividade.copy()
25             atividade.pop('respostas', None)
26         return jsonify(atividade)
27     except atividade_model.AtividadeNotFound:
28         return jsonify({'erro': 'Atividade não encontrada'}), 404
29
30 @atividade_bp.route('/', methods=['POST'])
31 def criar_atividade():
32     data = request.get_json()
33     id_disciplina = data.get('id_disciplina')
34     enunciado = data.get('enunciado')
35     respostas = data.get('respostas', [])
36
37     nova_atividade = atividade_model.criar_atividade(id_disciplina, enunciado, respostas)
38     return jsonify(nova_atividade), 201
39
40 @atividade_bp.route('/<int:id_atividade>', methods=['PUT'])
41 def atualizar_atividade(id_atividade):
42     data = request.get_json()
43     id_disciplina = data.get('id_disciplina')
```

**Postman Interface (Right Window):**

The Postman interface shows a collection named "Atividade Service Testes" with a test named "Obter Atividade para Professor". The test is a GET request to the endpoint `localhost:5002/atvidades/2/professor/1`. The response is a 200 OK status with a body containing the following JSON:

```
1 {
2   "enunciado": "Crie um servidor que envia email em Flask",
3   "id_atividade": 2,
4   "id_disciplina": 1
5 }
```

The bottom status bar of the Postman window shows "Postbot" and "Run" buttons.