

Para a implementação do trabalho, foi escolhida a implementação por API no Python. Foi escolhida a ferramenta Flask para a criação da API.

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

Assim como no trabalho anterior, o projeto usa das classes Apicultor e Apiario em sua implementação. A classe Apiario foi modificada para acrescentar novas funções.

```
class Apicultor:
    def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade

    def str_apicultor(self):
        return "Nome: " + self.nome + " | Idade: " + str(self.idade)
```

```
class Apiario:
    def __init__(self, id, num_colmeias):
        self.id = id
        self.num_colmeias = num_colmeias
        self.apicultores = []

    def str_apiario(self):
        strg = ""
        strg += "Apiario " + str(self.id) + " com " + str(self.num_colmeias) + " colmeias"
        return strg

    def get_apic_numb(self):
        return len(self.apicultores)

    def get_apicultor(self, id):
        return self.apicultores[id].str_apicultor()

    def add_apicultores(self, apicultor):
        self.apicultores.append(apicultor)

    def set_id(self, id):
        self.id = id
```

```
lista = []
```

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

Todos os objetos “Apiario” foram armazenados em “lista”. O Flask foi iniciado na variável “app”.

Seis funções foram implementadas para a aplicação. Duas para a inserção, uma de apiários e outra de apicultores, duas para a consulta, de todos os apiários ou de um específico e duas para a exclusão, de um apiário ou de apicultor.

Funções de consulta (GET):

```
#GET obter todos os apiários e funcionários
@app.route('/apiario', methods=['GET'])
def consult_apiarios():
    new_list = {}
    for ap in lista:
        new_list[str(ap.id)] = ap.str_apiario()

        for numb in range(ap.get_apic_numb()):
            str_aux = ""
            str_aux += str(ap.id) + ', Funcionario ' + str(numb)
            new_list[str_aux] = ap.get_apicultor(numb)

    return jsonify(new_list)
```

```
#GET obter apiário específico
@app.route('/apiario/<int:id>', methods=['GET'])
def consult_apiarios_id(id):
    new_list = {}

    for ap in lista:
        if ap.id == id:
            new_list["Apiario"] = ap.str_apiario()

            for numb in range(ap.get_apic_numb()):
                str_aux = "Funcionario " + str(numb)
                new_list[str_aux] = ap.get_apicultor(numb)

    return jsonify(new_list)
```

Funções de inserção (POST):

```
#POST criar novo apiário
@app.route('/apiario', methods=['POST'])
def add_apiario():
    ap_data = request.get_json()
    novo_ap = Apiario(ap_data["id"], ap_data["num_colmeias"])
    lista.append(novo_ap)

    new_list = {}
    for ap in lista:
        new_list[str(ap.id)] = ap.str_apiario()
    return jsonify(new_list)
```

```

#POST criar apicultor específico
@app.route('/apiario/<int:id>', methods=['POST'])
def add_apicultor(id):
    ap_data = request.get_json()
    novo_ap = Apicultor(ap_data["nome"], ap_data["idade"])

    new_list = {}

    new_list["Nome"] = ap_data["nome"]
    new_list["Idade"] = ap_data["idade"]

    for ap in lista:
        if ap.id == id:
            ap.add_apicultores(novo_ap)

    return jsonify(new_list)

```

Funções de exclusão (DELETE):

```

#DELETE apaga um apiário
@app.route('/apiario/<int:id>', methods=['DELETE'])
def delete_apiario(id):

    new_list = {}

    for ap in lista:
        if ap.id == id:
            lista.remove(ap)
        else:
            new_list[str(ap.id)] = ap.str_apiario()

    return jsonify(new_list)

```

```

#DELETE apaga um apicultor
@app.route('/apiario/<int:id>/<int:id2>', methods=['DELETE'])
def delete_apicultor(id,id2):

    new_list = {}

    for ap in lista:
        if ap.id == id:
            del ap.apicultores[id2]
            new_list["Apiario"] = ap.str_apiario()

            for numb in range(ap.get_apic_numb()):
                str_aux = "Funcionario " + str(numb)
                new_list[str_aux] = ap.get_apicultor(numb)

    return jsonify(new_list)

```

Por fim, a API é iniciada no localhost, porta 5000:

```
app.run(port=5000,host='localhost',debug=True)
```