

Simulação Chamada API

```
In [5]: import pandas as pd
import numpy as np
import requests
import json
```

```
In [6]: df_principal = pd.read_csv('../data/clientes.csv', sep=';')
df_principal.shape
```

```
Out[6]: (3, 13)
```

Criando o primeiro dataset - simulando o formulário

```
In [7]: class Pessoa:
    def __init__(self, nome, sobrenome, email, cep):
        self.nome = nome
        self.sobrenome = sobrenome
        self.email = email
        self.cep = cep

    def busca(self):
        response = requests.get(f'http://www.viacep.com.br/ws/{self.cep}/json/')
        # validando cep
        if response.status_code == 200:
            dados_json = json.loads(response.text)
            df = pd.DataFrame(data = dados_json, index=[0])
        else:
            print('Cep inválido !')
        return df

pessoa1 = Pessoa('rafa', 'silva', 'rafa@com', '25922000')
pessoa2 = Pessoa('teclado', 'sem fio', 'tec@com', '25922000')

dicio = {
    'nome': pessoa1.nome,
    'sobrenome': pessoa1.sobrenome,
    'email': pessoa1.email,
}
```

```
df1 = pd.DataFrame(data= dicio, index=[0])
df1
```

```
Out[7]:
```

	nome	sobrenome	email
0	rafa	silva	rafa@com

Criando o segundo dataset - simulando a API

```
In [8]: df2 = pessoa1.busca()
```

```
In [9]: # teste para adicionar novas linhas
b2 = pessoa2.busca()
df2.loc[len(df2)] = b2.values[0]
b2
```

```
Out[9]:
```

	cep	logradouro	complemento	bairro	localidade	uf	ibge	gia	ddd	siafi
0	25922-000	Rua J	Parque Itinga (Suruí)	Magé	RJ	3302502			21	5849

```
In [10]: b2.loc[len(b2)] = df2.values[0]
b2
```

```
Out[10]:
```

	cep	logradouro	complemento	bairro	localidade	uf	ibge	gia	ddd	siafi
0	25922-000	Rua J	Parque Itinga (Suruí)	Magé	RJ	3302502			21	5849
1	25922-000	Rua J	Parque Itinga (Suruí)	Magé	RJ	3302502			21	5849

Criei dois dataframes, pretendo agora concatená-los e ter apenas um dataframe.

Para isso vou utilizar a função **concat ([objeto_da_esquerda, objeto_da_direita], axis = 1)**

Passar o método axis=1 para juntar o índice das linhas eo ignore_index=True para sobrescrever valores null

```
In [11]: result = pd.concat([df1, df2], axis=1, ignore_index=True)
result
```

```
Out[11]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	rafa	silva	rafa@com	25922-000	Rua J	Parque Itinga (Suruí)	Magé	RJ	3302502		21	5849	
1	NaN	NaN	NaN	25922-000	Rua J	Parque Itinga (Suruí)	Magé	RJ	3302502		21	5849	

Obtem nomes colunas do df_principal

```
In [12]: columns_name = []
for i in df_principal:
    columns_name.append(i)
result.columns = columns_name
```

```
In [13]: for i in result.columns:
          print (i)
```

```
nome
sobrenome
email
cep
logradouro
complemento
bairro
localidade
uf
ibge
gia
ddd
siafi
```

Aqui finalizo a simulação

```
In [14]: df_principal = pd.concat([df_principal, result])
df_principal
```

```
Out[14]:
```

	nome	sobrenome	email	cep	logradouro	complemento	bairro	localidade	uf	ibge	gia	ddd	siafi
0	Rodrigo	Pires	rodrigo@email.com	1001000	Praça da Sé	lado ímpar	Sé	São Paulo	SP	3550308	1004.0	11	7107

	nome	sobrenome	email	cep	logradouro	complemento	bairro	localidade	uf	ibge	gia	ddd	siafi
1	Guilherme	Fernandes	guilherme@email.com	1001010	Rua Filipe de Oliveira	NaN	Sé	São Paulo	SP	3550308	1004.0	11	7107
2	Rafa	Silva	rafa@com	25922-000	Rua J	NaN	Parque Itinga (Suruí)	Magé	RJ	3302502	NaN	21	5849
0	rafa	silva	rafa@com	25922-000	Rua J		Parque Itinga (Suruí)	Magé	RJ	3302502		21	5849
1	NaN	NaN	NaN	25922-000	Rua J		Parque Itinga (Suruí)	Magé	RJ	3302502		21	5849

```
In [15]: response = requests.get(f'http://www.viacep.com.br/ws/2592300/json/')
if response.status_code == 200:
    print('Tudo ok')
else:
    print('Cep inválido !')
```

Cep inválido !

```
In [16]: a= df1.to_json(orient='index')
a
```

```
Out[16]: '{"0":{"nome":"rafa","sobrenome":"silva","email":"rafa@com"}}'
```

```
In [17]: class Busca:
def __init__(self, pp):
    self.pp = pp

def pesquisa(self):
    resp = requests.get(f'http://www.viacep.com.br/ws/{self.pp}/json/')
    busca = json.loads(resp.text)
    return busca.items()
dicionario = ''
```

```
In [19]: a = '25922100'
b = Busca(a)
```

```
b = Busca.pesquisa(b)
Busca.dicionario = b
```

```
In [20]: for i, v in Busca.dicionario : print(f'{i}: {v}' )
```

```
cep: 25922-100
logradouro: Estrada do Partido
complemento:
bairro: Vila Nova (Suruí)
localidade: Magé
uf: RJ
ibge: 3302502
gia:
ddd: 21
siafi: 5849
```

```
In [21]: class Busca:
        def __init__(self, pp):
            self.pp = pp

        def pesquisa(self):
            resp = requests.get(f'http://www.viacep.com.br/ws/{self.pp}/json/')
            if resp.status_code == 200:
                busca = json.loads(resp.text)
                print('encontrado')
                return busca
            else:
                print('erro na chamada')
```

```
In [22]: a = 25922160
        b = Busca(a)
        b = Busca.pesquisa(b)
```

encontrado

```
In [41]: resp = requests.get(f'http://www.viacep.com.br/ws/15121123402300/json/')
        print(resp.status_code)
```

400

```
In [ ]:
```

```
b['bairro']
```

In [152...

```
# TESTES DE VALIDAÇÃO
class Teste:
    def __init__(self, cep):
        self.cep = cep

    def api(self):
        response = requests.get(f'http://www.viacep.com.br/ws/{self.cep}/json/')
        if response.status_code == requests.codes.ok:
            dados_json = json.loads(response.text)
            df = pd.DataFrame(data=dados_json, index=[0])
            return df
        else:
            return response

aa = 2592222000
a = Teste(aa)
```

In [153...

```
a.cep
```

Out[153... 2592222000

In [154...

```
try:
    bb = a.api()
except:
    print('deu errado')
```

In [155...

```
bb
```

Out[155... <Response [400]>