

# CodeChallenge\_DataEngineer

---

## Propósito

Este desafio tem como objetivo avaliar a forma de resolver problemas do candidato. Portanto, a estrutura e a qualidade do código, bem como as tecnologias e técnicas utilizadas vão dizer muito sobre a metodologia utilizada.

Este desafio avaliará as seguintes habilidades:

- Entendimento e processamento de séries temporais financeiras.
- Eficiência no armazenamento dos dados
- Construção de uma API REST que responda às perguntas-chave, definidas nas especificações abaixo

## O Desafio

### 1º Trabalho

- 1 ) Processar o arquivo retirado do site do Tesouro Nacional.
- 2 ) Armazenar as informações obtidas da melhor forma possível.

### 2º Trabalho

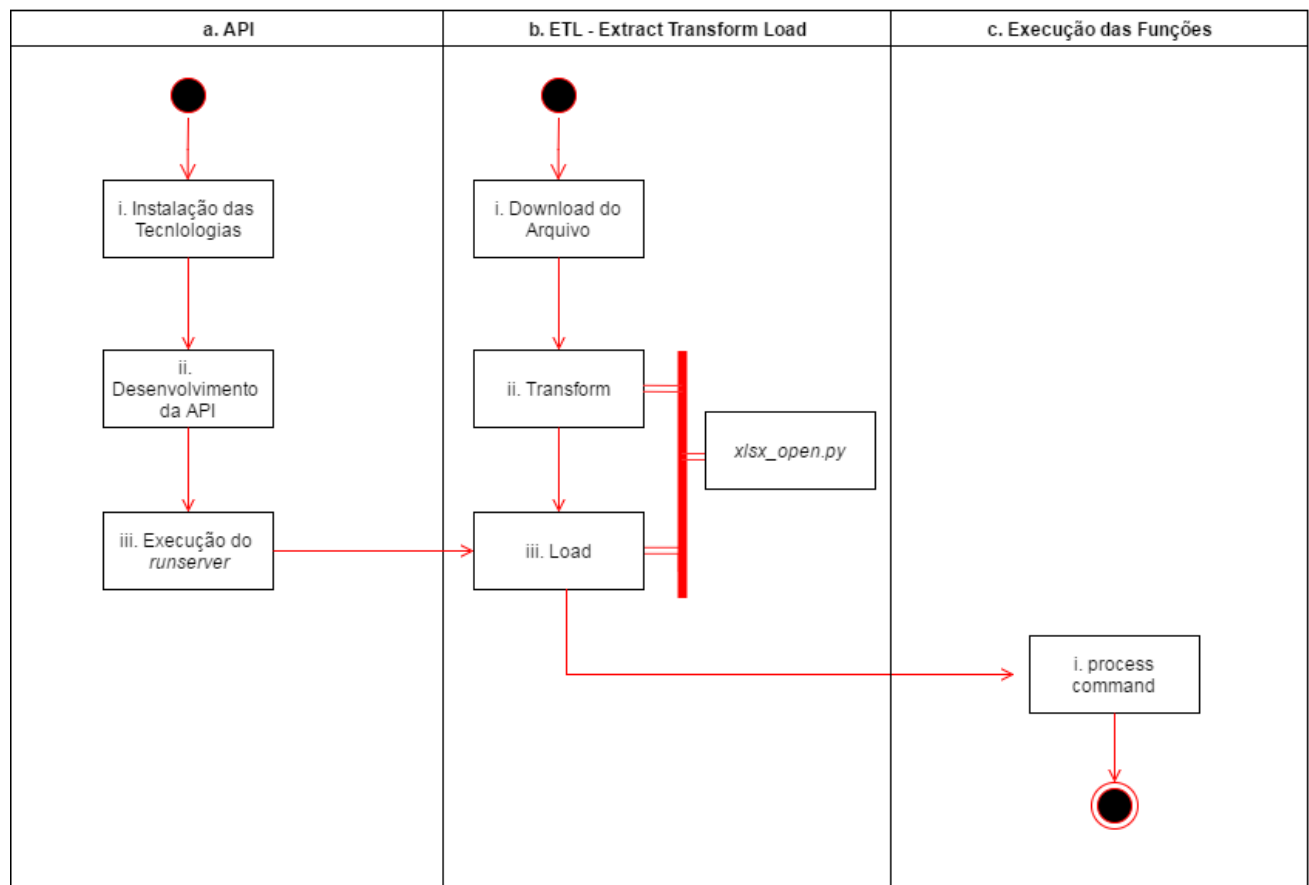
- 1) Construir uma API REST que execute as funções:
  - Adicionar um valor monetário a um título
  - Remover um valor monetário de um título
  - Atualizar um valor monetário para um título específico
  - Mostrar o histórico de um título específico
  - Comparar o histórico de dois ou mais títulos
  - Buscar valores de venda de um título em determinado período
  - Buscar valores de resgate de um título em determinado período

## Solução Proposta

1. Tecnologias utilizadas:

- a) GitHub (Social Code Hosting) - [https://github.com/luisffreitas/CodeChallenge\\_DataEngineer](https://github.com/luisffreitas/CodeChallenge_DataEngineer)
- b) Python
- c) Django (Web Framework)
- d) SQLite (Database)

## 2. Diagrama da solução:



### a. API

#### i. Instalação das Tecnologias

`pip install Django`

`pip install httpie`

`python startproject EasyInvest` - Cria-se um projeto chamado “EasyInvest”

`python manage.py startapp titulo_tesouro` - Cria-se um app “titulo\_tesouro”

Em “*Easyinvest/setting.py*” adicionar em `INSTALLED_APPS`:

`'rest_framework',`

`'titulo_tesouro',`

## ii. Desenvolvimento da API

### 1. Classe *Models.py*

```
models.py
1  from django.db import models
2
3
4  #criação da classe titulo_tesouro
5  class titulo_tesouro(models.Model):
6      #atributos dessa classe
7      created = models.DateTimeField(auto_now_add=True)
8      categoria_titulo = models.CharField(max_length=58)
9      mes = models.IntegerField()
10     ano = models.IntegerField()
11     acao = models.CharField(max_length=7)
12     valor = models.FloatField()
13
14     class Meta:
15         ordering = ('created',)
```

### 2. Classe *Serializer.py*

```
serializers.py
1  from rest_framework import serializers
2  from django.db import models
3  #importa a classe titulo_tesouro de models.py
4  from titulo_tesouro.models import titulo_tesouro
5
6  #criação da classe de serialização da classe titulo_tesouro
7  class titulo_tesouro_serializer(serializers.ModelSerializer):
8      class Meta:
9          model = titulo_tesouro
10         fields = ('id', 'created', 'categoria_titulo', 'mes', 'ano', 'acao', 'valor')
11
12     #validação dos dados para criação de um novo objeto
13     def create(self, validated_data):
14         return titulo_tesouro.objects.create(**validated_data)
15
16     #validação dos dados para atualização do objeto
17     def update(self, instance, validated_data):
18         instance.valor = validated_data.get('valor', instance.valor)
19         instance.save()
20         return instance
21
```

### 3. Classe *Urls.py* – Declaração das URL referentes a “titulo\_tesouro”

```
urls.py
1  from django.conf.urls import url
2  from titulo_tesouro import views
3  from rest_framework.urlpatterns import format_suffix_patterns
4
5  urlpatterns = [
6      url(r'^titulo_tesouro/$', views.titulo_tesouro_list),
7      url(r'^titulo_tesouro/(?P<pk>[0-9]+)/$', views.titulo_tesouro_detail),
8  ]
9
10 urlpatterns = format_suffix_patterns(urlpatterns)
```

#### 4. Classe Views.py

```
views.py
1 from rest_framework import status
2 from rest_framework.decorators import api_view
3 from rest_framework.response import Response
4 from titulo_tesouro.models import titulo_tesouro
5 from titulo_tesouro.serializers import titulo_tesouro_serializer
6
7 #API para GET (todos os objetos salvos) e POST (para criação de um novo objeto)
8 @api_view(['GET', 'POST'])
9 def titulo_tesouro_list(request, format=None):
10     if request.method == 'GET':
11         titulo = titulo_tesouro.objects.all()
12         serializer = titulo_tesouro_serializer(titulo, many=True)
13         #Retorna a lista de objetos titulo_tesouro serializados
14         return Response(serializer.data)
15
16     elif request.method == 'POST':
17         serializer = titulo_tesouro_serializer(data=request.data)
18         #Validação dados
19         if serializer.is_valid():
20             serializer.save()
21             return Response("Titulo criado com sucesso.")
22         return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
23
24 #API para GET (objetos id), UPDATE (objetos id) e DELETE (objetos id)
25 @api_view(['GET', 'UPDATE', 'DELETE'])
26 def titulo_tesouro_detail(request, pk, format=None):
27     #event handles caso o id(pk) não exista
28     try:
29         titulo = titulo_tesouro.objects.get(pk=pk)
30     except titulo_tesouro.DoesNotExist:
31         return Response(status=status.HTTP_404_NOT_FOUND)
32
33     if request.method == 'GET':
34         serializer = titulo_tesouro_serializer(titulo)
35         #Retorna as informações do objeto serializado
36         return Response(serializer.data)
37
38     elif request.method == 'UPDATE':
39         serializer = titulo_tesouro_serializer(titulo, data=request.data)
40         if serializer.is_valid():
41             #Atualiza as informações caso elas sejam validas
42             serializer.save()
43             return Response("Titulo atualizado com sucesso.")
44         return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
45
46     elif request.method == 'DELETE':
47         #deleta objeto id
48         titulo.delete()
49         return Response("Titulo deletado")
```

#### iii. Execução do Servidor

python manage.py makemigrations titulo\_tesouro

python manage.py migrate

python manage.py runserver

#### b. ETL

##### i. Download do Arquivo

Salvar como C:\Series\_Temporais\_Tesouro\_Direto.xlsx

ii. e iii. - xlsx\_open.py

```

1  from openpyxl.reader.excel import load_workbook
2  import sqlite3
3  from datetime import datetime
4
5  # Conectando com o BD e Criando a TABLE
6  db = sqlite3.connect('EasyInvest/db.sqlite3')
7  cur = db.cursor()
8
9  #comando para criar tabela caso não exista, desenvolvido a partir de Models.py
10 cur.execute('''CREATE TABLE IF NOT EXISTS titulo_tesouro_titulo_tesouro(
11      id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
12      created DATETIME NOT NULL,
13      categoria_titulo VARCHAR (58) NOT NULL,
14      mes INTEGER NOT NULL,
15      ano INTEGER NOT NULL,
16      acao VARCHAR (7) NOT NULL,
17      valor REAL NOT NULL
18  )''')
19
20 #Abrir arquivo de series temporais
21 wb = load_workbook(filename = 'C:\Series_Temporais_Tesouro_Direto.xlsx')
22 ws = wb['Planilha1']
23
24 #declaração de variáveis auxiliares
25 i = 0 #variavel utilizada para identificar inicio dos valores das series temporais
26 mes = ""
27 ano = ""
28 nome_Serie = [] #vetor utilizado para registrar nome da serie temporal
29 nome_Serie.append("blank")
30 nome_Serie.append("blank")
31
32 #python lê célula a célula do arquivo
33 for row in ws.rows:
34     for cell in row:
35         i = i + 1
36         valor_celula = str(cell.value)
37         created = str(datetime.now())
38         #registra nome da series no vetor
39         if "Nome da série" in valor_celula:
40             nome_Serie.append(valor_celula)
41         #apos a contagem de 155 células comecam os valores correspondentes as Series
42
43         #a primeira informação registrada é o mês e ano
44         if (i > 155) and i % 14 == 2 :
45             mes = valor_celula[5:-12]
46             ano = valor_celula[:4]
47
48         #em seguida os valores para cada serie são registrado
49         #primeiras 6 células - series de Vendas
50         if (i > 155) and i % 14 > 2 and i % 14 <= 8 :
51             nome_SerieTemp = nome_Serie[i%14-1]
52             categoria_titulo = nome_SerieTemp[40:]
53             acao = "Vendas"
54             valor = valor_celula
55
56             #registramos os valores das series temporais no banco de dados SQLite3 criado
57             cur.execute('''INSERT INTO titulo_tesouro_titulo_tesouro(created, categoria_titulo, mes, ano, acao,valor)
58                 VALUES(?,?,?,?,?)''', (created,categoria_titulo,mes, ano, acao,valor))
59
60         #ultimas 6 células - series de Resgates
61         elif (i > 155) and (i % 14 > 8 or i % 14 == 0):
62             if(i % 14 == 0):
63                 nome_SerieTemp = nome_Serie[13]
64             else:
65                 nome_SerieTemp = nome_Serie[i%14-1]
66             categoria_titulo = nome_SerieTemp[42:]
67             acao = "Resgates"
68             valor = valor_celula
69
70             #registramos os valores das series temporais no banco de dados SQLite3 criado
71             cur.execute('''INSERT INTO titulo_tesouro_titulo_tesouro(created, categoria_titulo, mes, ano, acao,valor)
72                 VALUES(?,?,?,?,?)''', (created,categoria_titulo,mes, ano, acao,valor))
73
74 #Commit das execuções
75 db.commit()
76
77 =====

```

### c. Execução das Funções

- Adicionar um valor monetário a um título

```
C:\Windows\system32\cmd.exe

C:\>http --json POST http://127.0.0.1:8000/titulo_tesouro/ acao=Vendas
valor=101 ano=2017 categoria_titulo=LTN mes=01
HTTP/1.0 200 OK
Allow: OPTIONS, GET, POST
Content-Length: 28
Content-Type: application/json
Date: Sun, 09 Apr 2017 21:40:14 GMT
Server: WSGIServer/0.2 CPython/3.6.1
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN

"Titulo criado com sucesso."
```

id	created	categoria_titulo	mes	ano	acao	valor
1461	2017-04-09 13:44:27.618529	NTN-B	2	2016	Resgates	128.89
1462	2017-04-09 13:44:27.618529	NTN-B Principal	2	2016	Resgates	91.75
1463	2017-04-09 13:44:27.618529	NTN-C	2	2016	Resgates	1.42
1464	2017-04-09 13:44:27.618529	NTN-F	2	2016	Resgates	16.28
1465	2017-04-09 13:44:27.618529	LTN	3	2016	Vendas	194.74
1466	2017-04-09 13:44:27.618529	LFT	3	2016	Vendas	472.68
1467	2017-04-09 13:44:27.618529	NTN-B	3	2016	Vendas	179.27
1468	2017-04-09 13:44:27.618529	NTN-B Principal	3	2016	Vendas	871.77
1469	2017-04-09 13:44:27.618529	NTN-C	3	2016	Vendas	0
1470	2017-04-09 13:44:27.618529	NTN-F	3	2016	Vendas	32.05
1471	2017-04-09 13:44:27.618529	LTN	3	2016	Resgates	103.25
1472	2017-04-09 13:44:27.618529	LFT	3	2016	Resgates	243.08
1473	2017-04-09 13:44:27.618529	NTN-B	3	2016	Resgates	74.86
1474	2017-04-09 13:44:27.618529	NTN-B Principal	3	2016	Resgates	303.05
1475	2017-04-09 13:44:27.618529	NTN-C	3	2016	Resgates	0.04
1476	2017-04-09 13:44:27.618529	NTN-F	3	2016	Resgates	23.7
1477	2017-04-09 13:44:27.618529	LTN	4	2016	Vendas	149.39
1478	2017-04-09 13:44:27.618529	LFT	4	2016	Vendas	363.95
1479	2017-04-09 13:44:27.618529	NTN-B	4	2016	Vendas	134.2
1480	2017-04-09 13:44:27.618529	NTN-B Principal	4	2016	Vendas	723.78
1481	2017-04-09 13:44:27.618529	NTN-C	4	2016	Vendas	0
1482	2017-04-09 13:44:27.618529	NTN-F	4	2016	Vendas	17
1483	2017-04-09 13:44:27.618529	LTN	4	2016	Resgates	77.05
1484	2017-04-09 13:44:27.618529	LFT	4	2016	Resgates	178.35
1485	2017-04-09 13:44:27.618529	NTN-B	4	2016	Resgates	56.9
1486	2017-04-09 13:44:27.618529	NTN-B Principal	4	2016	Resgates	173.81
1487	2017-04-09 13:44:27.618529	NTN-C	4	2016	Resgates	0.93
1488	2017-04-09 13:44:27.618529	NTN-F	4	2016	Resgates	17.15
1489	2017-04-09 21:39:38.283457	LTN	9	2017	Vendas	101
1490	2017-04-09 21:40:14.364521	LTN	1	2017	Vendas	101

- Remover um valor monetário de um título

SQLiteStudio (3.1.1) - [titulo\_tesouro\_titulo\_tesouro (db)]

Database Structure View Tools Help

Databases Filter by name

db (SQLite 3)

Tables (11)

- auth\_group
- auth\_group\_permissions
- auth\_permission
- auth\_user
- auth\_user\_groups
- auth\_user\_permissions
- django\_admin\_log
- django\_content\_type
- django\_migrations
- django\_session
- titulo\_tesouro\_titulo\_tesouro

Views

Grid view Form view

Filter data Total rows loaded: 1490

	id	created	categoria_titulo	mes	ano	acao	valor
1461	1461	2017-04-09 13:44:27.618529	NTN-B	2	2016	Resgates	128.89
1462	1462	2017-04-09 13:44:27.618529	NTN-B Principal	2	2016	Resgates	91.75
1463	1463	2017-04-09 13:44:27.618529	NTN-C	2	2016	Resgates	16.28
1464	1464	2017-04-09 13:44:27.618529	NTN-F	2	2016	Resgates	16.28
1465	1465	2017-04-09 13:44:27.618529	LTN	3	2016	Vendas	194.74
1466	1466	2017-04-09 13:44:27.618529	LFT	3	2016	Vendas	472.68
1467	1467	2017-04-09 13:44:27.618529	NTN-B	3	2016	Vendas	179.27
1468	1468	2017-04-09 13:44:27.618529	NTN-B Principal	3	2016	Vendas	871.77
1469	1469	2017-04-09 13:44:27.618529	NTN-C	3	2016	Vendas	0
1470	1470	2017-04-09 13:44:27.618529	NTN-F	3	2016	Vendas	32.05
1471	1471	2017-04-09 13:44:27.618529	LTN	3	2016	Resgates	103.25
1472	1472	2017-04-09 13:44:27.618529	LFT	3	2016	Resgates	243.08
1473	1473	2017-04-09 13:44:27.618529	NTN-B	3	2016	Resgates	74.86
1474	1474	2017-04-09 13:44:27.618529	NTN-B Principal	3	2016	Resgates	303.05
1475	1475	2017-04-09 13:44:27.618529	NTN-C	3	2016	Resgates	0.04
1476	1476	2017-04-09 13:44:27.618529	NTN-F	3	2016	Resgates	23.7
1477	1477	2017-04-09 13:44:27.618529	LTN	4	2016	Vendas	149.39
1478	1478	2017-04-09 13:44:27.618529	LFT	4	2016	Vendas	363.95
1479	1479	2017-04-09 13:44:27.618529	NTN-B	4	2016	Vendas	134.2
1480	1480	2017-04-09 13:44:27.618529	NTN-B Principal	4	2016	Vendas	723.78
1481	1481	2017-04-09 13:44:27.618529	NTN-C	4	2016	Vendas	0
1482	1482	2017-04-09 13:44:27.618529	NTN-F	4	2016	Vendas	17
1483	1483	2017-04-09 13:44:27.618529	LTN	4	2016	Resgates	77.05
1484	1484	2017-04-09 13:44:27.618529	LFT	4	2016	Resgates	178.35
1485	1485	2017-04-09 13:44:27.618529	NTN-B	4	2016	Resgates	56.9
1486	1486	2017-04-09 13:44:27.618529	NTN-B Principal	4	2016	Resgates	173.81
1487	1487	2017-04-09 13:44:27.618529	NTN-C	4	2016	Resgates	0.93
1488	1488	2017-04-09 13:44:27.618529	NTN-F	4	2016	Resgates	17.15
1489	1491	2017-04-09 21:39:38.283457	LTN	9	2017	Vendas	101
1490	1492	2017-04-09 21:40:14.364521	LTN	1	2017	Vendas	101

titulo\_tesouro\_titulo\_tesouro (db)

```
C:\>http --json DELETE http://127.0.0.1:8000/titulo_tesouro/1488/
HTTP/1.0 200 OK
Allow: DELETE, OPTIONS, GET, UPDATE
Content-Length: 17
Content-Type: application/json
Date: Sun, 09 Apr 2017 21:44:35 GMT
Server: WSGIServer/0.2 CPython/3.6.1
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN

"Titulo deletado"
```

SQLiteStudio (3.1.1) - [titulo\_tesouro\_titulo\_tesouro (db)]

Database Structure View Tools Help

Databases Filter by name

db (SQLite 3)

Tables (11)

- auth\_group
- auth\_group\_permissions
- auth\_permission
- auth\_user
- auth\_user\_groups
- auth\_user\_permissions
- django\_admin\_log
- django\_content\_type
- django\_migrations
- django\_session
- titulo\_tesouro\_titulo\_tesouro

Views

Grid view Form view

Filter data Total rows loaded: 1489

	id	created	categoria_titulo	mes	ano	acao	valor
1460	1460	2017-04-09 13:44:27.618529	LFT	2	2016	Resgates	149.67
1461	1461	2017-04-09 13:44:27.618529	NTN-B	2	2016	Resgates	128.89
1462	1462	2017-04-09 13:44:27.618529	NTN-B Principal	2	2016	Resgates	91.75
1463	1463	2017-04-09 13:44:27.618529	NTN-C	2	2016	Resgates	16.28
1464	1464	2017-04-09 13:44:27.618529	NTN-F	2	2016	Resgates	16.28
1465	1465	2017-04-09 13:44:27.618529	LTN	3	2016	Vendas	194.74
1466	1466	2017-04-09 13:44:27.618529	LFT	3	2016	Vendas	472.68
1467	1467	2017-04-09 13:44:27.618529	NTN-B	3	2016	Vendas	179.27
1468	1468	2017-04-09 13:44:27.618529	NTN-B Principal	3	2016	Vendas	871.77
1469	1469	2017-04-09 13:44:27.618529	NTN-C	3	2016	Vendas	0
1470	1470	2017-04-09 13:44:27.618529	NTN-F	3	2016	Vendas	32.05
1471	1471	2017-04-09 13:44:27.618529	LTN	3	2016	Resgates	103.25
1472	1472	2017-04-09 13:44:27.618529	LFT	3	2016	Resgates	243.08
1473	1473	2017-04-09 13:44:27.618529	NTN-B	3	2016	Resgates	74.86
1474	1474	2017-04-09 13:44:27.618529	NTN-B Principal	3	2016	Resgates	303.05
1475	1475	2017-04-09 13:44:27.618529	NTN-C	3	2016	Resgates	0.04
1476	1476	2017-04-09 13:44:27.618529	NTN-F	3	2016	Resgates	23.7
1477	1477	2017-04-09 13:44:27.618529	LTN	4	2016	Vendas	149.39
1478	1478	2017-04-09 13:44:27.618529	LFT	4	2016	Vendas	363.95
1479	1479	2017-04-09 13:44:27.618529	NTN-B	4	2016	Vendas	134.2
1480	1480	2017-04-09 13:44:27.618529	NTN-B Principal	4	2016	Vendas	723.78
1481	1481	2017-04-09 13:44:27.618529	NTN-C	4	2016	Vendas	0
1482	1482	2017-04-09 13:44:27.618529	NTN-F	4	2016	Vendas	17
1483	1483	2017-04-09 13:44:27.618529	LTN	4	2016	Resgates	77.05
1484	1484	2017-04-09 13:44:27.618529	LFT	4	2016	Resgates	178.35
1485	1485	2017-04-09 13:44:27.618529	NTN-B	4	2016	Resgates	56.9
1486	1486	2017-04-09 13:44:27.618529	NTN-B Principal	4	2016	Resgates	173.81
1487	1487	2017-04-09 13:44:27.618529	NTN-C	4	2016	Resgates	0.93
1488	1491	2017-04-09 21:39:38.283457	LTN	9	2017	Vendas	101
1489	1492	2017-04-09 21:40:14.364521	LTN	1	2017	Vendas	101

titulo\_tesouro\_titulo\_tesouro (db)

- Atualizar um valor monetário para um título específico

The image shows a workflow for updating a database record via an API. The top window is SQLiteStudio (3.11.1) showing a table named 'titulo\_tesouro' with columns: id, created, categoria\_titulo, mes, ano, acao, and valor. The table contains 30 records. The bottom window is a terminal showing a REST client request to update the 'acao' field of a specific record.

**SQLiteStudio Table Data (Initial State):**

id	created	categoria_titulo	mes	ano	acao	valor
1	2017-04-09 13:44:27.571729	LTN	9	2017	Vendas	101
2	2017-04-09 13:44:27.571729	LFT	1	2006	Vendas	25.43
3	2017-04-09 13:44:27.571729	NTN-B	1	2006	Vendas	20.79
4	2017-04-09 13:44:27.571729	NTN-B Principal	1	2006	Vendas	5.26
5	2017-04-09 13:44:27.571729	NTN-C	1	2006	Vendas	2.39
6	2017-04-09 13:44:27.571729	NTN-F	1	2006	Vendas	3.18
7	2017-04-09 13:44:27.571729	LTN	1	2006	Resgates	91.1
8	2017-04-09 13:44:27.571729	LFT	1	2006	Resgates	35.56
9	2017-04-09 13:44:27.571729	NTN-B	1	2006	Resgates	0.87
10	2017-04-09 13:44:27.571729	NTN-B Principal	1	2006	Resgates	0.04
11	2017-04-09 13:44:27.571729	NTN-C	1	2006	Resgates	6.71
12	2017-04-09 13:44:27.571729	NTN-F	1	2006	Resgates	2.29
13	2017-04-09 13:44:27.571729	LTN	2	2006	Vendas	15.49
14	2017-04-09 13:44:27.571729	LFT	2	2006	Vendas	10.47
15	2017-04-09 13:44:27.571729	NTN-B	2	2006	Vendas	14.56
16	2017-04-09 13:44:27.571729	NTN-B Principal	2	2006	Vendas	16.03
17	2017-04-09 13:44:27.571729	NTN-C	2	2006	Vendas	18.28
18	2017-04-09 13:44:27.571729	NTN-F	2	2006	Vendas	2.17
19	2017-04-09 13:44:27.571729	LTN	2	2006	Resgates	6.23
20	2017-04-09 13:44:27.571729	LFT	2	2006	Resgates	4.52
21	2017-04-09 13:44:27.571729	NTN-B	2	2006	Resgates	5.09
22	2017-04-09 13:44:27.571729	NTN-B Principal	2	2006	Resgates	0.83
23	2017-04-09 13:44:27.571729	NTN-C	2	2006	Resgates	3.3
24	2017-04-09 13:44:27.571729	NTN-F	2	2006	Resgates	0.48
25	2017-04-09 13:44:27.571729	LTN	3	2006	Vendas	17.4
26	2017-04-09 13:44:27.571729	LFT	3	2006	Vendas	10.44
27	2017-04-09 13:44:27.571729	NTN-B	3	2006	Vendas	14.94
28	2017-04-09 13:44:27.571729	NTN-B Principal	3	2006	Vendas	6.84
29	2017-04-09 13:44:27.571729	NTN-C	3	2006	Vendas	2.3
30	2017-04-09 13:44:27.571729	NTN-F	3	2006	Vendas	2.28

**Terminal Output:**

```
C:\>http --json UPDATE http://127.0.0.1:8000/titulo_tesouro/1/
acao=Resgate valor=111 ano=1111 categoria_titulo=NTN-C mes=11
HTTP/1.0 200 OK
Allow: DELETE, OPTIONS, GET, UPDATE
Content-Length: 32
Content-Type: application/json
Date: Sun, 09 Apr 2017 21:48:51 GMT
Server: WSGIServer/0.2 CPython/3.6.1
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN

"Titulo atualizado com sucesso."
```

**SQLiteStudio Table Data (After Update):**

id	created	categoria_titulo	mes	ano	acao	valor
1	2017-04-09 13:44:27.571729	NTN-C	11	1111	Resgate	111
2	2017-04-09 13:44:27.571729	LFT	1	2006	Vendas	25.43
3	2017-04-09 13:44:27.571729	NTN-B	1	2006	Vendas	20.79
4	2017-04-09 13:44:27.571729	NTN-B Principal	1	2006	Vendas	5.26
5	2017-04-09 13:44:27.571729	NTN-C	1	2006	Vendas	2.39
6	2017-04-09 13:44:27.571729	NTN-F	1	2006	Vendas	3.18
7	2017-04-09 13:44:27.571729	LTN	1	2006	Resgates	91.1
8	2017-04-09 13:44:27.571729	LFT	1	2006	Resgates	35.56
9	2017-04-09 13:44:27.571729	NTN-B	1	2006	Resgates	0.87
10	2017-04-09 13:44:27.571729	NTN-B Principal	1	2006	Resgates	0.04
11	2017-04-09 13:44:27.571729	NTN-C	1	2006	Resgates	6.71
12	2017-04-09 13:44:27.571729	NTN-F	1	2006	Resgates	2.29
13	2017-04-09 13:44:27.571729	LTN	2	2006	Vendas	15.49
14	2017-04-09 13:44:27.571729	LFT	2	2006	Vendas	10.47
15	2017-04-09 13:44:27.571729	NTN-B	2	2006	Vendas	14.56
16	2017-04-09 13:44:27.571729	NTN-B Principal	2	2006	Vendas	16.03
17	2017-04-09 13:44:27.571729	NTN-C	2	2006	Vendas	18.28
18	2017-04-09 13:44:27.571729	NTN-F	2	2006	Vendas	2.17
19	2017-04-09 13:44:27.571729	LTN	2	2006	Resgates	6.23
20	2017-04-09 13:44:27.571729	LFT	2	2006	Resgates	4.52
21	2017-04-09 13:44:27.571729	NTN-B	2	2006	Resgates	5.09
22	2017-04-09 13:44:27.571729	NTN-B Principal	2	2006	Resgates	0.83
23	2017-04-09 13:44:27.571729	NTN-C	2	2006	Resgates	3.3
24	2017-04-09 13:44:27.571729	NTN-F	2	2006	Resgates	0.48
25	2017-04-09 13:44:27.571729	LTN	3	2006	Vendas	17.4
26	2017-04-09 13:44:27.571729	LFT	3	2006	Vendas	10.44
27	2017-04-09 13:44:27.571729	NTN-B	3	2006	Vendas	14.94
28	2017-04-09 13:44:27.571729	NTN-B Principal	3	2006	Vendas	6.84
29	2017-04-09 13:44:27.571729	NTN-C	3	2006	Vendas	2.3
30	2017-04-09 13:44:27.571729	NTN-F	3	2006	Vendas	2.28