

Exercícios – Sprint 2

Neste projeto, foi extraído dados da API do site Educação Inteligente [1]. Os dados foram extraídos e armazenados localmente, para que os exercícios da sprint foquem sua atenção apenas no processamento das estruturas de dados.

[1] <http://educacao.dadosabertosbr.com/api>

FASE 1

1) Desenvolver a função **extrair_chaves_dicionario**. Esta função recebe como parâmetro um dicionário. Retorna uma lista. Cada elemento da lista possui a chave do dicionário passado como parâmetro.

2) Desenvolver a função **extrair_tuplas_cidades**. Esta função recebe como parâmetro uma lista strings e o código do estado. Cada elemento (string) possui o número id da cidade, dois pontos (:) e o nome da cidade, conforme exemplo a figura 1:

```
['3500907:ALTAIR',  
'3501004:ALTINOPOLIS',  
'3501103:ALTO ALEGRE',  
'3501152:ALUMINIO',  
'3501202:ALVARES FLORENCE',  
'3501301:ALVARES MACHADO',  
'3501400:ALVARO DE CARVALHO',  
'3501509:ALVINLANDIA']
```

Figura 1

```
[('3500907', 'ALTAIR', 'SP'),  
( '3501004', 'ALTINOPOLIS', 'SP'),  
( '3501103', 'ALTO ALEGRE', 'SP'),  
( '3501152', 'ALUMINIO', 'SP'),  
( '3501202', 'ALVARES FLORENCE', 'SP'),  
( '3501301', 'ALVARES MACHADO', 'SP'),  
( '3501400', 'ALVARO DE CARVALHO', 'SP'),  
( '3501509', 'ALVINLANDIA', 'SP')]
```

Figura 2

Retorna uma lista de tuplas com os registros no formato: código da cidade, nome da cidade e nome do estado, conforme demonstrado na figura 2.

FASE 2

Neste momento, espera-se que você tenha uma lista de tuplas, com as cidades (lista_cidades). O consumo de API consome um tempo considerável. Considerando este cenário, os objetos **dados_escolares_id** e **dados_escolares_nome** estão previamente carregados.

3) Desenvolver a função **extrair_dados_escolares_por_municipio**. Esta função recebe como parâmetro um dicionário de dados (**dados_escolares_id** e **dados_escolares_nome**) e um parâmetro, o código da cidade pesquisada ou nome da cidade. Caso o código do município ou cidade não seja encontrado, retornar um dicionário vazio.

Retorna uma lista com os dados do município pesquisado. O primeiro campo da lista será um número inteiro, com o número de escolas encontradas. O segundo parâmetro será uma lista de dicionários. Cada dicionário corresponde a uma escola do município pesquisado.

4) Desenvolver a função **extrair_dados_municipio_codigo_nome**. Esta função recebe como parâmetro o retorno da função **extrair_dados_escolares_por_municipio** (exercício 3).

Retorna uma lista de tuplas com os campos: cod e nome, conforme modelo a seguir:

```
In [3]: extrair_dados_municipio_codigo_nome(araras)  
Out[3]:  
[(35246141, 'ADALGISA PERIM BALESTRO FRANZINI PROFA EMEF'),  
(35019926, 'ALBERTO FERES PREFEITO ETE'),  
(35085029, 'ANTONIA MARQUES DAHMEN EMEF'),  
(35092400, 'ANTONIO SEVERINO EMEIEF'),  
(35145427, 'APAE DE ARARAS'),  
(35374180, 'AQUARELA ESCOLA DE EDUCACAO INFANTIL'),  
(35478465, 'AQUARELA ESCOLA DE EDUCACAO INFANTIL'),  
(35808295, 'ARARAS JUNIOR LICEU'),  
(35801793, 'ARARAS LICEU'),
```

5) Desenvolver a função **extrair_dados_municipio_codigo_nome_situacaoFuncionamento**. Esta função recebe como parâmetro o retorno da função **extrair_dados_escolares_por_municipio** (exercício 3). Retorna uma lista de tuplas com os campos: cod , nome e situacaoFuncionamentoTxt, conforme modelo a seguir:

```
In [1]: extrair_dados_municipio_codigo_nome_situacaoFuncionamento(araras)
Out[1]:
[(35246141, 'ADALGISA PERIM BALESTRO FRANZINI PROFA EMEF', 'Em atividade'),
 (35019926, 'ALBERTO FERES PREFEITO ETE', 'Em atividade'),
 (35085029, 'ANTONIA MARQUES DAHMEN EMEF', 'Em atividade'),
 (35092400, 'ANTONIO SEVERINO EMEIEF', 'Em atividade'),
 (35145427, 'APAE DE ARARAS', 'Em atividade'),
 (35374180, 'AQUARELA ESCOLA DE EDUCACAO INFANTIL', 'Em atividade'),
 (35478465, 'AQUARELA ESCOLA DE EDUCACAO INFANTIL', 'Em atividade'),
 (35808295, 'ARARAS JUNIOR LICEU', 'Em atividade'),
 (35801793, 'ARARAS LICEU', 'Em atividade'),
```

6) Desenvolver a função

extrair_dados_municipio_codigo_nome_situacaoFuncionamento_dependenciaAdministrativa.

Esta função recebe como parâmetro o retorno da função **extrair_dados_escolares_por_municipio** (exercício 3).

Retorna uma lista de tuplas com os campos:

- cod
- nome
- situacaoFuncionamentoTxt
- dependenciaAdministrativaTxt

Conforme modelo a seguir:

```
In [1]: extrair_dados_municipio_codigo_nome_situacaoFuncionamento_dependenciaAdministrativa(araras)
...: ras)
Out[1]:
[(35246141,
 'ADALGISA PERIM BALESTRO FRANZINI PROFA EMEF',
 'Em atividade',
 'Municipal'),
 (35019926, 'ALBERTO FERES PREFEITO ETE', 'Em atividade', 'Estadual'),
 (35085029, 'ANTONIA MARQUES DAHMEN EMEF', 'Em atividade', 'Municipal'),
 (35092400, 'ANTONIO SEVERINO EMEIEF', 'Em atividade', 'Municipal'),
 (35145427, 'APAE DE ARARAS', 'Em atividade', 'Privada'),
 (35374180, 'AQUARELA ESCOLA DE EDUCACAO INFANTIL', 'Em atividade', 'Privada'),
 (35478465, 'AQUARELA ESCOLA DE EDUCACAO INFANTIL', 'Em atividade', 'Privada'),
 (35808295, 'ARARAS JUNIOR LICEU', 'Em atividade', 'Privada'),
 (35801793, 'ARARAS LICEU', 'Em atividade', 'Privada'),
```

7) Desenvolver a função **extrair_tupla_dados_municipio**. Esta função recebe como parâmetro o retorno da função **extrair_dados_escolares_por_municipio** (exercício 3) e strings (uma ou mais strings) com o(s) nome(s) dos campos procurados, via args. Caso o campo não exista, ignorar o campo passado.

Retorna uma lista de tuplas. Cada tupla deve conter os dados da escola, conforme o exemplo a seguir:

```
In [2]: extrair_tupla_dados_municipio(araras, 'nome', 'cidade')
Out[2]:
[('ADALGISA PERIM BALESTRO FRANZINI PROFA EMEF', 'ARARAS'),
 ('ALBERTO FERES PREFEITO ETE', 'ARARAS'),
 ('ANTONIA MARQUES DAHMEN EMEF', 'ARARAS'),
 ('ANTONIO SEVERINO EMEIEF', 'ARARAS'),
 ('APAE DE ARARAS', 'ARARAS'),
 ('AQUARELA ESCOLA DE EDUCACAO INFANTIL', 'ARARAS'),
 ('AQUARELA ESCOLA DE EDUCACAO INFANTIL', 'ARARAS'),
 ('ARARAS JUNIOR LICEU', 'ARARAS'),
 ('ARARAS LICEU', 'ARARAS'),
```

8) Desenvolver a função **gravar_dados_em_CSV**. Esta função recebe como parâmetros uma string com o nome do arquivo em CSV, uma lista ou tupla com os campos e uma lista de tuplas, com os dados desejados.

FASE 3

Neste momento, espera-se que você tenha desenvolvido todas as funções propostas anteriormente e todos os testes unitários estejam em ordem. Parabéns !

Descomentar os trechos com comentários. Faça seus próprios testes. Com uso do interpretador Python, faça testes com as funções desenvolvidas e responda aos exercícios propostos no ambiente virtual moodle.