

Exercícios – Sprint 2

Neste projeto, foi gerado um conjunto de arquivos CSV. Cada arquivo possui registros de tentativas de compras suspeita de fraude. Cada tentativa de fraude está registrada contém as seguintes informações:

Estrutura da tupla:

- Nome
- CPF
- e-mail
- data/hora
- hostname
- IP
- Dados do navegador
- Endereço
- Número do cartão de crédito
- Mês/ano que expira o cartão de crédito

1) Desenvolver a função **carregar_arquivo_para_lista**. Esta função recebe como parâmetro o nome de um arquivo CSV com tentativas de fraude e retorna uma lista de tuplas com os registros presentes no arquivo.

2) Desenvolver a função **carregar_arquivos_para_lista**. Esta função recebe como parâmetro o nome ou uma tupla de nomes de arquivo(s) CSV com tentativas de fraude e retorna uma lista de tuplas com os registros presentes no(s) arquivo(s).

3) Desenvolver a função **busca_mes_vencimento_cartao_credito**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas) e em kwargs. As chaves esperadas deste kwargs podem ser mes e ano. Caso as chaves de kwargs não serem passadas, assumir como mês=03 e ano=23

Esta função retorna a lista com todos registros que possuam o mês/ano que expira o cartão de crédito passado como parâmetro.

4) Desenvolver a função **contar_ocorrencias_cartao_credito_por_mes**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas). Realizar a busca no campo Mês/ano que expira o cartão de crédito.

Esta função retorna um dicionário em que a chave é o mês do vencimento do cartão de crédito e o valor é o total de tentativas de fraudes.

5) Desenvolver a função **contar_ocorrencias_cartao_credito_por_ano**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas). Realizar a busca no campo Mês/ano que expira o cartão de crédito.

Esta função retorna um dicionário em que a chave é o ano do vencimento do cartão de crédito e o valor é o total de tentativas de fraudes.

6) Desenvolver a função **busca_cvc**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas) e um código de verificação do cartão de crédito (CVC).

Esta função retorna a lista com todos registros que possuam no cartão de crédito o código de verificação informado. Caso nenhum código de verificação seja informado, considerar o valor padrão “043”.

7) Desenvolver a função **busca_lista_cvc**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas) , uma string com CVC ou uma tupla de strings com CVCs.

Esta função retorna a lista com todos registros que possuam no cartão de crédito o(s) código(s) de verificação informado. Caso nenhum código de verificação seja informado, considerar o valor padrão “043”.

8) Desenvolver a função **contar_ocorrencia_por_estado**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna um dicionário em que a chave é o estado, presente no endereço e o valor é o total de tentativas de fraudes.

9) Desenvolver a função **busca_dados_navegador**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas) e uma string com dados do navegador utilizado na tentativa de fraude.

Esta função retorna a lista com todos registros com dados do navegador passado. Caso nenhum código de verificação seja informado, considerar o valor padrão "Chrome/24".

10) Desenvolver a função **contar_ocorrencia_por_sufixo dominio**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna um dicionário em que a chave é o sufixo do domínio informado (.br, .com, .net), presente no campo *hostname* e o valor é o total de tentativas de fraudes.

11) Desenvolver a função **contar_ocorrencia_por dominio_email**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna um dicionário em que a chave é o sufixo do domínio do e-mail informado (uol.com.br, google.com, etc), presente no campo *email* e o valor é o total de tentativas de fraudes.

12) Desenvolver a função **buscar dominio_mais_utilizado**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna uma tupla. Cada o primeiro elemento da tupla deve ser o domínio mais utilizado e o segundo elemento o número de ocorrência, conforme o modelo:

('ig.com.br', 27)

13) Desenvolver a função **contar_ocorrencia_por_semana**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna uma lista de tuplas. Cada tupla possui o nome do dia e o número de ocorrência, conforme o modelo:

[('Segunda', 40), ('Terça', 28), ('Quarta', 37), ('Quinta', 26), ('Sexta', 44), ('Sábado', 37), ('Domingo', 38)]

14) Desenvolver a função **contar_ocorrencia_por_hora**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna um dicionário em que a chave é a hora informada, presente no campo *data/hora* e o valor é o total de tentativas de fraudes.

15) Desenvolver a função **contar_ocorrencia_por_mes**. Esta função recebe como parâmetro a lista de tentativas de compras suspeitas de fraude (lista de tuplas).

Esta função retorna um dicionário em que a chave é o mês informado, presente no campo *data/hora* e o valor é o total de tentativas de fraudes.

Orientação a objetos

Neste projeto, iremos migrar o conjunto de funções para uma única classe.

Em ambos os projetos, foi criado registros de tentativas de compras suspeita de fraude. Cada tentativa de fraude está registrada contém as seguintes informações:

Estrutura da tupla:

- Nome
- CPF
- e-mail
- data/hora
- hostname
- IP
- Dados do navegador
- Endereço
- Número do cartão de crédito
- Mês/ano que expira o cartão de crédito

16) Desenvolver a classe **Sprint2**.

Esta classe deve instanciar uma lista vazia interna chamada *lista*.

17) Desenvolver o método **carregar_arquivo_para_lista**. Esta função recebe como parâmetro o nome de um arquivo CSV com tentativas de fraude. A lista de tuplas com os registros presentes no arquivo deve ser armazenada no atributo *lista*.

Não deve ser armazenado registros duplicados.

18) Desenvolver o método **busca_trecho_nome**. Este método recebe uma string com o nome ou trecho do procurado.

Este método retorna uma lista de tuplas. Cada tupla contém somente o nome e o CPF de pessoas que possuam o trecho do nome em seu nome.

19) Desenvolver o método **busca_email**. Este método recebe como parâmetro uma string com o e-mail procurado. Este método retorna a tupla com o registro caso encontrado. Caso o e-mail procurado não esteja na lista, retorna uma tupla vazia.

20) Desenvolver o método **lista_estados**. Esta função não possui passagem de parâmetros e retorna a lista com todos os estados em que exista registro de tentativa de fraude. Não deve retornar registros duplicados.

21) Desenvolver o método **busca_mes_vencimento_cartao_credito**. Este método recebe como parâmetro uma string com o mês que expira o cartão de crédito. Esta função retorna a lista com todos registros que possuam o mês/ano que expira o cartão de crédito passado como parâmetro.

22) Desenvolver o método **busca_prefixo_sufixo_ip**. Este método recebe como parâmetro uma string com o prefixo do IP procurado e uma string com o sufixo do IP procurado.

Esta função retorna a lista com todos registros que possuam o IP com o prefixo passado como parâmetro e com o sufixo passado como parâmetro.

23) Desenvolver o método **contar_ocorrencia_por_dominio_hostname**. Este método recebe um ou mais domínios como parâmetro.

Este método retorna uma lista de tuplas. Cada tupla deve conter o domínio passado e o número de ocorrências registradas no campo hostname.

24) Desenvolver o método **contar_ocorrencia_por_dominio_email**. Este método recebe um ou mais domínios como parâmetro.

Este método retorna uma lista de tuplas. Cada tupla deve conter o domínio passado e o número de ocorrências registradas no campo e-mail.

25) Desenvolver o método **busca_mes_aniversario**. Este método recebe como parâmetro o mês procurado (entre 1 e 12) e retorna uma lista de tuplas. Cada tupla contém o registro de ocorrência que possui o mês no campo data/hora.

26) Descomentar o trecho da função main para testar os métodos **relatorio_lista_para_txt** e **relatorio_lista_para_csv**. No final do código sprint2_OO.py