Um ficheiro CSV (Comma-separated values) é um tipo de ficheiro de texto usado regularmente em Informática para representar dados de forma estruturada. Estes ficheiros são usados por serem facilmente lidos por programas informáticos. Abaixo temos o exemplo do conteúdo de um ficheiro CSV a que vamos dar o nome de "*compras.csv*":

produto,preço unitário,quantidade  
maçãs,0.50€,10  
garrafas de água,0.33€,6  
vassoura,15€,1  
  
Num ficheiro com o nome **epp4.py** deverá codificar as funções abaixo.  
  
Função **processa\_csv(nome\_ficheiro) [10 valores]**  
Escreva uma função em Python chamada **processa\_csv** que recebe como argumento uma cadeia de caracteres **nome\_ficheiro**. Esta função abre o ficheiro cujo nome foi passado como argumento e que se assume que é um ficheiro CSV, lê o conteúdo do ficheiro e devolve uma lista em que cada elemento é um dicionário correspondente a uma linha do ficheiro CSV. Os dicionários terão como chaves os títulos das colunas e como valores as cadeias de caracteres correspondentes a cada coluna para a linha em questão. Assume-se que a primeira linha do ficheiro CSV corresponde sempre ao cabeçalho do ficheiro, ou seja, aos títulos dados a cada uma das colunas. O exemplo abaixo usa o ficheiro "*compras.csv*" definido acima. A função deve suportar um ficheiro CSV arbitrário. Apenas é necessário considerar ficheiros CSV em que o separador entre valores é uma vírgula (**','**). Não é necessário suportar casos em que uma vírgula aparece nos valores. Deve ser possível suportar casos em que o valor é vazio (e.g. **maçãs,,10**).  
  
Sugestão: A função **split** de uma cadeia de caracteres permite partir a cadeia de caracteres utilizando um caracter separador, que é passado como argumento. Por exemplo, dada uma cadeia de caracteres **"a b c"**, **"a b c".split(' ')** devolve a lista **["a", "b", "c"]**.  
  
Exemplo  
 **>>> processa\_csv('compras.csv')  
[{'produto': 'maçãs', 'preço unitário': '0.50€', 'quantidade': '10'}, {'produto': 'garrafas de água', 'preço unitário': '0.33€', 'quantidade': '6'}, {'produto': 'vassoura', 'preço unitário': '15€', 'quantidade': '1'}]**

Função **gera\_csv(dados, nome\_ficheiro) [10 valores]**

Escreva uma função em Python, chamada **gera\_csv**,que recebe como argumentos uma lista de dicionários **dados**(com a mesma estrutura do resultado da função **processa\_csv**)e uma cadeia de caracteres **nome\_ficheiro**, e cria um ficheiro CSV com o nome que foi passado como argumento. A primeira linha do ficheiro deverá corresponder ao cabeçalho que deverá ser gerado a partir das chaves dos dicionários da lista **dados**.Todos os elementos da lista **dados**devem ter o mesmo conjunto de chaves. Caso um dos elementos não satisfaça esta condição, deverá ser lançada uma exceção do tipo **ValueError**, com a mensagem "**dados incoerentes**". Nesse caso, não deverá ser gerado nenhum ficheiro. As linhas seguintes do ficheiro devem corresponder aos valores de cada elemento da lista **dados** separados por uma vírgula ("**,**")e ordenados de acordo com a ordem definida no cabeçalho**.**A ordem das colunas no cabeçalho é arbitrária.  
  
**Exemplo**  
  
**>>> gera\_csv([{'produto': 'maçãs', 'preço unitário': '0.50€', 'quantidade': '10'}, {'produto': 'garrafas de água', 'preço unitário': '0.33€', 'quantidade': '6'}, {'produto': 'vassoura', 'preço unitário': '15€', 'quantidade': '1'}], "compras2.csv")**  
**>>>**  
  
=== Conteúdo do ficheiro compras2.csv ===  
**preço unitário,produto,quantidade  
0.50€,maçãs,10  
0.33€,garrafas de água,6  
15€,vassoura,1**============================  
  
**>>> gera\_csv([{'produto': 'maçãs', 'preço unitário': '0.50€'}, {'produto': 'garrafas de água', 'preço unitário': '0.33€', 'quantidade': '6'}, {'produto': 'vassoura', 'preço unitário': '15€', 'quantidade': '1'}], "compras2.csv")  
...  
ValueError: dados incoerentes**

Notas Importantes

* Encontra-se em anexo o ficheiro **compras.csv**que poderá utilizar para testar o seu código. Este ficheiro está codificado em UTF-8.
* Os valores apresentados nos exemplos são apenas exemplificativos: as suas funções devem aceitar quaisquer valores para os respetivos parâmetros, dentro do tipo e gama especificados.
* O programa pedido deverá ser submetido num ficheiro com o nome **epp4.py**. Na primeira linha do ficheiro deverá constar, em comentário, o seu nome e número de aluno.

Critérios de Avaliação

Cada uma das funções será classificada individualmente, para a cotação indicada, de acordo com os seguintes critérios.

* Resultados corretos: 70%
  + *Se a função consegue produzir o resultado esperado, incluíndo, caso seja pedido, se consegue produzir as exceções corretas para parâmetros de entrada inválidos.*
* Qualidade do código: 30%
  + *Se o código da função está corretamente comentado (não em excesso), se está bem estruturado e se é de fácil compreensão.*