2. **(4 valores)** Escreva uma função em Python que lhe permita calcular o produto de duas matrizes. Para o efeito, considere que uma matriz é representada por uma lista de listas em que cada lista interna representa uma linha da matriz. A função **`produto(matriz1, matriz2)`** deverá retornar uma matriz se o produto for possível e **`None`** caso contrário.

Exemplo 1:

$$[1 \quad 2] \times \begin{bmatrix} 3 \\ 4 \end{bmatrix} = [11]$$

```
produto([[1,2]], [[3],[4]]) -> [[11]]
```

Exemplo 2:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

```
produto([[1,2],[3,4]], [[5,6],[7,8]]) -> [[19, 22], [43, 50]])
```

Exemplo 3 (não é possível multiplicar):

$$[1 \quad 2] \times \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = \ !$$

```
produto([[1,2]], [[3],[4], [5]]) -> None
```

Fundamentos de Programação

Avaliação Final 2

2016/2017

Objectivos:

- Programação em Python
- Estruturas de Dados
- Ordenação e Pesquisa

Duração

- Deverá completar os exercícios propostos em 2h00

Instruções

- Faça login com o username **sessao1** e a password **um**.
- No Desktop encontra sete ficheiros (F2_1.py, F2_2.py, pauta.csv, p1.csv, p2.csv, p3.csv, p4.csv).
- Deve editar **F2_1.py** e **F2_2.py** para responder aos exercícios.
- Não altere os nomes dos ficheiros.
- No final, feche todas as janelas e faça **logout**, mas não desligue o PC.

1. **(16 valores)** Corrigir exames de programação é uma tarefa longa e trabalhosa. Para facilitar o processo, os professores testam automaticamente os programas dos alunos e geram um ficheiro de texto (CSV) por cada pergunta. Esses ficheiros contêm linhas no formato:

   NMEC, Compila, testeA, testeB, testeC

   onde **NMEC** é o número mecanográfico do aluno, **Compila** é um valor booleano (True/False) e **testeA**, **testeB** e **testeC** indicam o resultado de cada teste (-1: crasha, 0: resposta errada, 1: resposta certa).

   a. **(2 valores)** Faça uma função (chamada **`ler`**) que leia o conteúdo de um ficheiro de texto para uma estrutura de dados adequada.

   b. **(4 valores)** Cada pergunta é classificada de 0% a 100% segundo o número de testes que acerta:
      testeA: 40%
      testeB: 40%
      testeC: 20%
      Se o programa não compilar, a classificação é 0%, se algum dos testes fizer o programa crashar, existe uma penalização de 5% na classificação da pergunta.
      Crie uma função (chamada **`classificar`**) que, dado um tuplo (compila, testeA, testeB, testC), calcula a classificação correspondente.

   c. **(4 valores)** Um exame é composto por várias perguntas. Escreva uma função (chamada **`apurar`**) que leia ficheiros de várias perguntas e que peça ao utilizador o número de valores de cada pergunta (ver exemplo). No final a função deverá devolver um dicionário com NMEC como chave e a nota do exame como valor.

   d. **(6 valores)** Finalmente é necessário imprimir a lista de alunos e notas respectivas.
      Implemente uma função (chamada **`imprimir`**) que receba um dicionário com NMEC e nota do exame e imprima:
      i. **(4 valores)** Listagem de alunos ordenada por nota (decrescente). É fornecido o ficheiro **pauta.csv**, que contém o NMEC e Nome dos alunos.
      ii. **(2 valores)** Estatísticas:
         1. Número de alunos Aprovados (nota >= 9.5)
         2. Número de alunos Reprovados
         3. Percentagem de Reprovações

No final o seu programa deverá produzir o seguinte output:

```
Ficheiro (ENTER para terminar)? p1.csv
Valores? 3
Ficheiro (ENTER para terminar)? p2.csv
Valores? 7
Ficheiro (ENTER para terminar)? p3.csv
Valores? 6
Ficheiro (ENTER para terminar)? p4.csv
Valores? 4
Ficheiro (ENTER para terminar)?

82735 Luisa Marques                                    18.6
58743 Filipe Vilela                                    15.4
80543 Bernardo Vilela                                  15.2
29761 Sonia Abreu                                      13.2
66023 Joana Melo                                       13.1
79981 Julio Morais                                     12.6
30009 Alexandra Bastos                                 11.5
87882 Pedro Oliveira                                   11.0
55019 Carlos Pascoal                                   10.6
 2121 Daniela Seabra                                    8.6
Aprovados: 9
Reprovados: 1
Percentagem Reprovados: 10.0%
```