

Histogramas & Diagramas

(versão completa)

Objectivos

Este primeiro guião tem dois objectivos principais: aprender a usar o juiz automático (o Mooshak) com base na resolução de um problema e iniciar a aquisição de competências de interpretação de problemas de programação, com vista à selecção de um desenho de classes apropriado. Cada uma das seguintes duas partes irá atacar, separadamente, cada um desses problemas.

1ª Parte – Histogramas et al.

O problema

Muito resumidamente, pretende-se calcular algumas estatísticas e desenhar histogramas com os resultados de POO nos anos lectivos transactos. Os ficheiros de entrada correspondem a pautas (ex: exame de recurso de 2007/2008) e estão em formato *tab-delimited* (nº do aluno na 1ª coluna e nota a POO na 2ª). Estes ficheiros de entrada são fornecidos neste guião inicial, na página da disciplina. Em conjunto com os correspondentes ficheiros de saída (que só se fornecem, neste guião, para um dos ficheiros de entrada), constituem a bateria de testes que o Mooshak usa para verificar se um dado programa é ou não válido. Nos guiões seguintes poderemos, ou não, disponibilizar os ficheiros de teste, total ou parcialmente, conforme seja adequado. **A data limite para a submissão no Mooshak é dia 9 de Março às 23h55m GMT.**

Como programar de forma a poder usar o Mooshak?

1. No Eclipse fazer File -> New -> Project.
2. Escolher Java Project.
3. No Project layout escolher "Create separate source and output folders".
4. Finish.
5. No package explorer ir até à src do projecto criado.
6. Fazer New -> Package com nome poo.
7. Criar as classes necessárias na package poo.
8. A função main deverá estar na classe Main (é esta função que o Mooshak irá correr no arranque). Como esta 1ª parte do guião tem 3 tarefas, pode criar uma função para cada uma delas e vai-as descomentando uma de cada vez, como em baixo, em cada submissão ao Mooshak.

```
package poo;

public class Main
{
    public static void main(String[] args)
    {
        taskA();
        // taskB();
        // taskC();
    }
}
```

No *Mooshak* há vários ficheiros de teste de entrada, e os correspondentes de saída, para cada tarefa. O seu programa só será aceite se todos os testes passarem. Para isso, o que o seu programa escreve na consola deve ser rigorosamente igual ao que é pedido, nem um carácter a mais, nem um a menos. Os dados de entrada são lidos automaticamente da consola, como se alguém os estivesse a teclar na altura. Para testar o seu programa no Eclipse pode usar a janela de consola para introduzir valores de entrada um por um, ou então fazer lá paste de todo o ficheiro de dados de entrada que copiou deste enunciado. Em qualquer dos casos terá de instruir o computador que terminou a entrada de dados, introduzindo na consola o carácter de *EOF (End Of File)*. No *Windows* isso consegue-se fazendo *Control-Z* (carregando simultaneamente na tecla de “Ctrl” e na letra “z”).

Para evitar estar sempre a fazer paste do ficheiro de entrada na consola do Eclipse pode guardá-lo num ficheiro de dados. Sugere-se que crie um directório denominado `data` ao mesmo nível que a directoria `src` e lá dentro guarde os seus ficheiros de dados. Agora, em vez de usar a consola do Eclipse, deve usar a consola do sistema operativo (geralmente uma janela de fundo preto em que pode dar comandos em texto). Para correr o seu programa na consola do sistema operativo, deve redirigir o *input*, para obter os dados a partir do ficheiro de entrada. Se o mesmo se chamar `2008_Normal.txt` e estiver guardado na directoria `src`, então deverá escrever o seguinte comando: `java poo.Main < ..\data\2008_Normal.txt`. O programa escreverá o resultado na consola. Se quiser salvarguardar o *output*, pode fazê-lo redirigindo-o para um ficheiro, assim: `java poo.Main < ..\data\2008_Normal.txt > ..\data\taskA_out.txt`. Verifique a directoria `data`, e confira que o ficheiro `taskA_out.txt` contém o resultado pretendido.

Para efectuar leituras da consola pode usar a classe `java.util.Scanner` (veja [aqui a sua documentação](#)) com uma estrutura do seguinte tipo:

```
Scanner in = new Scanner(System.in);
```

```
while (in.hasNext())
{
    int x = in.nextInt();
    int y = in.nextInt();
    ...
}
```

Tome atenção que o último carácter dos ficheiros de entrada usados no *Mooshak* é o carácter `\n` (salto de linha), razão pela qual são visualizados num editor de texto como um ficheiro cuja última linha está vazia.

Tarefa A (Estatísticas descritivas)

Para começar a aquecer vamos fazer umas pequenas contas. Para cada uma das pautas fornecidas, pretende-se saber qual foi a nota que ocorreu mais vezes (moda), a média e o desvio padrão, em linhas sucessivas. Se houver empates na moda, será devolvido a que corresponda a uma nota maior. Os valores da média e do [desvio padrão](#) deverão ser representados com duas casas decimais. Atenção que nesta e nas restantes tarefas só iremos considerar resultados a que tenha sido atribuída uma nota numérica. Repare que há várias avaliações traduzidas numa letra (ex: A => aluno Ausente).

Por exemplo para o exame final em época normal de 2008 (pauta “2008_Normal”) o resultado é o seguinte:

14
10,11
4,63

Atenção: se o seu computador estiver configurado em língua inglesa, então provavelmente vão-lhe aparecer pontos em vez de vírgulas como separadores da parte decimal. Não se preocupe com isso, porque o Mooshak realizará automaticamente a comparação com o formato por omissão.

Tarefa B (Tabela de Frequências)

Um histograma é uma representação num gráfico de barras das frequências (n° de vezes que ocorre na amostra) de cada possível valor. Como aqui a amostra é um conjunto de notas na escala $[0, 20]$, cada barra indicará, respectivamente, o número de 20's, de 19's, de 18's e assim sucessivamente. Dado que ainda não sabe desenhar barras num programa em Java, iremos gerar ficheiros *tab-delimited* contendo o valor da nota na 1ª coluna e a sua frequência na 2ª coluna. Esses ficheiros de saída, que constituem aquilo a que os estatísticos chamam de [Tabela de Frequências](#), podem ser lidos por uma folha de cálculo como o Excel para gerar automaticamente um histograma. Pois bem, escreva um programa que, dada uma pauta, produza uma tabela de frequências das notas, em formato *tab-delimited*, ordenada por ordem decrescente do valor da nota (obs: o carácter de controlo `\t` permite inserir o espaçamento pretendido entre as duas colunas).

Por exemplo, para o exame final em época normal de 2008 (pauta “2008_Normal”) o resultado é o seguinte:

20	0
19	5
18	4
17	6
16	5
15	6
14	15
13	15
12	11
11	12
10	9
9	2
8	8
7	14
6	14
5	9
4	8
3	5
2	6
1	1
0	0

Importe o ficheiro de saída resultante no Excel, seleccione uma representação em gráfico de barras e mostre o histograma assim construído ao responsável pelo seu turno prático.

Tarefa C (Quadro de honra)

Pretende-se agora determinar, dada uma pauta, quais foram os dez melhores alunos (1ª coluna a nota, ordenada por ordem decrescente e 2ª coluna o número, ordenado por ordem decrescente dentro da nota). Se houver empates são escolhidos os alunos com número de aluno mais recentes (maiores). Só podem entrar para o quadro de honra alunos que tenham obtido uma nota de pelo menos 17 valores. Assim, por exemplo numa pauta de época especial, ou mesmo de recurso, podemos ter um quadro de honra incompleto (por exemplo só com 3 alunos) ou mesmo vazio. Neste último caso, o ficheiro de saída deverá conter apenas uma linha com 10 traços horizontais (10 sinais “menos”).

Por exemplo, para o exame final em época normal de 2008 (pauta “2008_Normal”) o resultado é o seguinte:

19	26470
19	25505
19	25186
19	25184
19	25120
18	26315
18	25518
18	25448
18	25148
17	25836

Registo no Mooshak

Em POO usaremos muito o Mooshak, para as submissões de guiões, problemas, etc. Para se registar no Mooshak, leia com atenção as seguintes regras:

1. Siga o [link de acesso ao Mooshak](#)
2. Escolha a opção 2. *Register for on-line contest*
3. Na página seguinte aparecerá a janela de registo com o aspecto seguidamente representado.

4. Escolha o concurso: POO 2008/09 (Guião 1)
5. Crie o seu *username* com o formato seguinte (não se esqueça dos *underscores*):

NºAluno_IdMoole_PrimeiroNome_UltimoApellido
(exemplo: 12345_5678_Joao_Pestana)

- a) O *NºAluno* é obviamente o seu número oficial que lhe foi atribuído aquando da inscrição na FCT;
 - b) O *idMoodle* é o seu identificador no Moodle, que pode obter na barra de endereço do seu browser quando lá consulta o seu perfil (ex: <http://moodle.fct.unl.pt/user/view.php?id=5678&course=2065>)
 - c) No primeiro nome e último apelido não use qualquer carácter acentuado (acentos, cedilhas,...) e comece cada palavra com uma Maiúscula e todos os outros caracteres em minúsculas. **Seja muito cuidadoso na inclusão do seu número e identificador do Moodle porque se não o fizer, não poderemos identificar de quem foram as submissões ou combinar os pontos que obtiverem no Mooshak com aqueles obtidos no Moodle. É você que fica a perder ☹**
6. Indique um endereço de *email* válido. Isto é muito importante, pois a *password* de acesso ao Mooshak ser-lhe-á enviada para este email. Note que alguns sistemas de *email* classificam esta mensagem como SPAM. Por isso, se não encontrar a mensagem com a *password* na mailbox procure nas pastas de SPAM. Não se esqueça de guardar cuidadosamente a *password*, pois esta será usada durante o semestre inteiro. (Por exemplo, guarde a mensagem numa pasta que não seja apagado, ou conserve-a numa mensagem no telemóvel, ou anote-a no cartão de estudante, ou ...)
 7. Seleccione o grupo dos Alunos (LEI) e carregue em *Submit*.
 8. E pronto, já está inscrito no Mooshak ☺

Submissão das soluções das tarefas no Mooshak

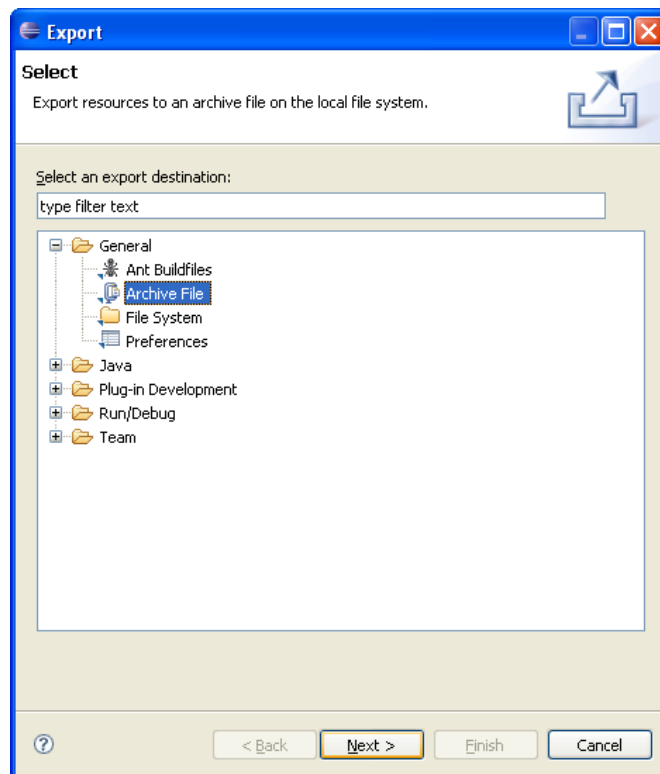
9. Usando o seu *username* e a *password* que recebeu por email, faça agora login ao Mooshak, usando o mesmo link que em cima, mas agora carregando no botão de Login. Na janela que lhe aparece escolha o mesmo concurso e introduza as suas credenciais, da forma representada a seguir.



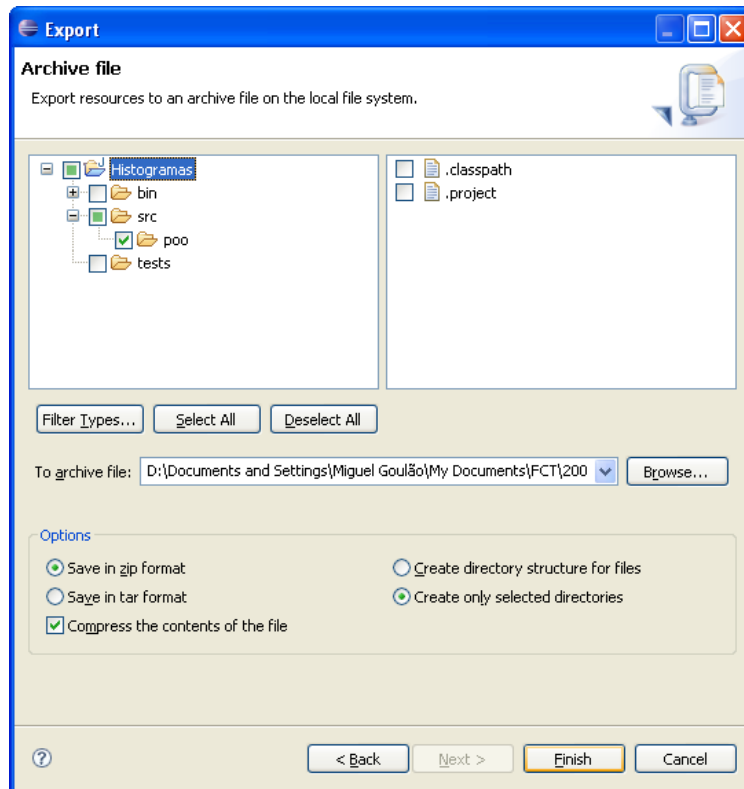
The image shows a Mooshak authentication form with a light blue background. At the top, it says "Mooshak authentication". Below that, there are three input fields: "Contest" with a dropdown menu showing "POO 2008/09 (Guião 1)", "User" with the text "12345_5678_Joao_Pest", and "Password" with a masked field of dots. At the bottom, there are four buttons: "Login", "Reset", "Register", and "Guest".

10. Agora que está registado, pode realizar as submissões das soluções das suas tarefas A, B e C ao Mooshak, da seguinte forma:

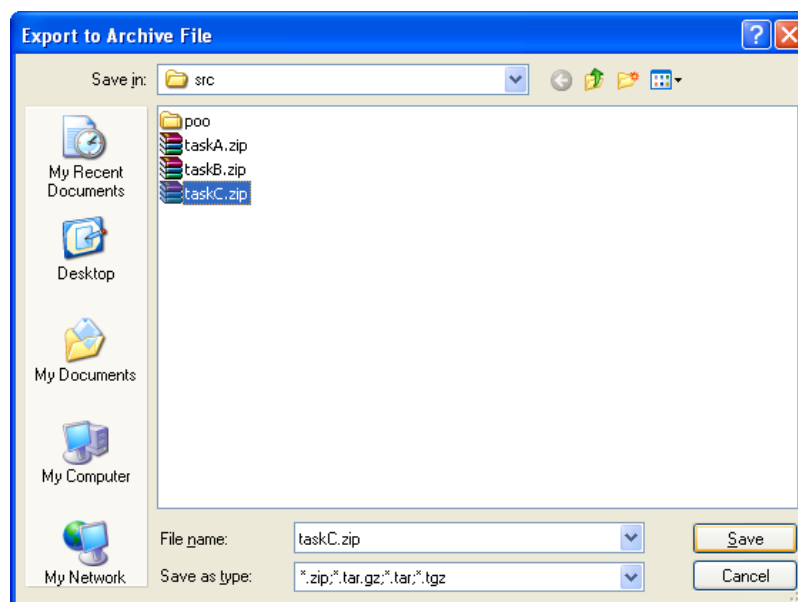
- a) Construa um zip contendo a sua solução para cada uma tarefa. Para tal seleccione no Eclipse o menu *File -> Export*



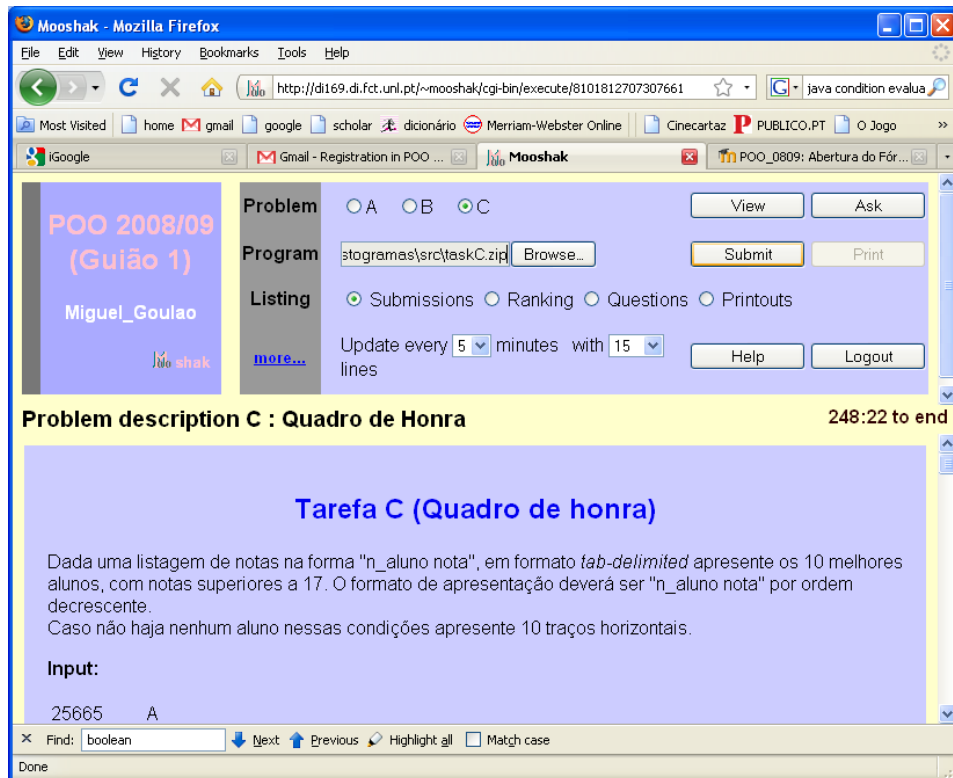
- b) Escolha *Archive File* e carregue em *Next >*. Seleccione a pasta *poo*, que contém todos os ficheiros, use o botão *Browse* para escolher o nome e localização do arquivo zip (mantendo as opções como se apresentam na imagem seguinte).



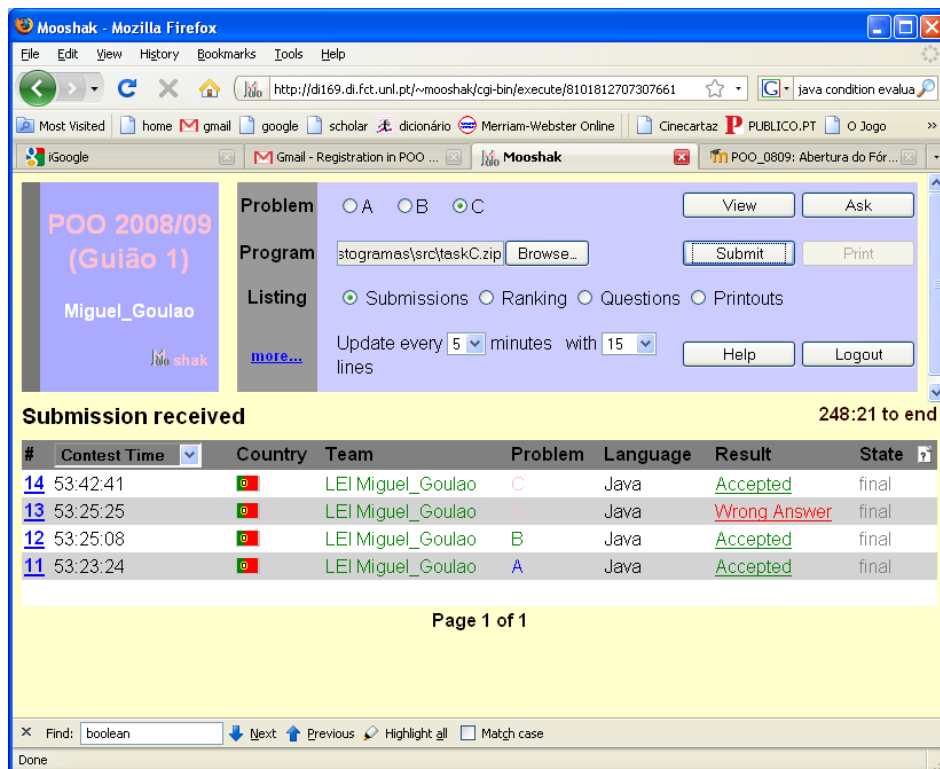
c) Guarde um zip diferente para cada tarefa.



d) No Mooshak escolha a tarefa em causa (A, B ou C) e submeta o ficheiro zip correspondente. Neste exemplo, vamos submeter o arquivo com a tarefa C.



- e) Faça submit e aguarde o resultado. Em caso de erro de compilação, clique no link "*Compile Time Error*". Se não houver concordância entre os ficheiros de saída gerados pela sua solução e os da bateria de testes então aparecerá "*Wrong Answer*". Na figura seguinte, a tarefa C foi aceite à segunda tentativa



2ª Parte – Diagramas

Nesta parte iremos propor um conjunto de pequenas especificações de requisitos (enunciados de problemas em língua natural), para os quais se pretende obter um desenho de classes apropriado. Para cada uma das especificações pretende-se que conduza as seguintes três tarefas. **Esta 2ª parte servirá de treino para um Microteste que será anunciado na aula teórica de 6 de Março.**

Tarefa A (Diagrama de classes)

Proponha um diagrama de classes em UML (Unified Modeling Language), a notação actualmente mais usada para este efeito, onde se identifique:

- quais as classes mais adequadas para representar o problema em causa;
- quais as responsabilidades concretas de cada classe;
- como deverão essas classes cooperar para atingir os objectivos pretendidos (Nota: não se esqueça das cardinalidades e da direcionalidade das associações);
- que alternativas existem relativamente aos 3 pontos anteriores.

Tarefa B (Atributos de qualidade)

Construa um quadro em que nas linhas representa cada um dos requisitos especificados (R1, R2, R2, ...) e nas colunas as alternativas identificadas na tarefa anterior (identificadas com A1, A2, A3, ...). Discuta, do ponto de vista da qualidade do produto, quais os prós e os contras de cada uma das alternativas. Em particular, considere as seguintes características de qualidade:

- EM – Eficiência do ponto de vista da memória central
- EP – Eficiência do ponto de vista do processador
- FM – Facilidade de manutenção
- FR – Facilidade de reutilização

Em cada célula do quadro supracitado coloque as siglas das características e um sufixo a indicar a importância relativa face ao requisito e a alternativa em causa. O significado dos sufixos é o representado na tabela seguinte.

SUFIXO	SIGNIFICADO
++	Favorece muito
+	Favorece apreciavelmente
+-	Nem favorece, nem penaliza
-	Penaliza apreciavelmente
--	Penaliza muito

Assim, se na célula {A3, R2} estiver **EP++**, isso significa que a alternativa A3 favorece muito a eficiência do ponto de vista do processador, para o caso do requisito R2.

Tarefa C (De UML para Java)

Forneça a implementação da estrutura de classes em Java de cada uma das alternativas identificadas na tarefa A.

ESPECIFICAÇÃO E1 – Facturação

Introdução:

As facturas são emitidas em nome de um cliente e contêm várias linhas, cada uma delas relativa à compra de uma dada quantidade de um produto. Os preços de venda dos produtos são fixos. Pretende-se saber:

- R1 – o número de facturas emitidas em nome de um dado cliente (adimensional)
 - R2 – o total de uma factura (euros)
 - R3 – a quantidade vendida de um dado produto (adimensional)
 - R4 – o volume de vendas de um dado produto (euros)
 - R5 – a quantidade vendida de um dado produto a um dado cliente (adimensional)
-

ESPECIFICAÇÃO E2 – Inscrições nas disciplinas

Introdução:

Os alunos inscrevem-se em mais de uma disciplina e obtêm resultados em 3 momentos: avaliação contínua, época normal e época recurso. A disciplina tem uma equipa docente constituída por um professor responsável e vários professores assistentes. Pretende-se saber:

- R1 – nº de alunos inscritos numa dada disciplina
 - R2 – disciplinas a que está inscrito um dado aluno
 - R3 – média obtida pelos alunos que realizaram a época normal numa disciplina
 - R4 – média actual de um dado aluno
 - R5 – número de alunos do total das disciplinas leccionadas por um dado professor
-

ESPECIFICAÇÃO E3 – Salão de casamentos

Introdução:

O salão de casamentos oferece vários menus à escolha. Cada menu está organizado em três secções – entradas, pratos e sobremesas – cada uma contendo vários itens. Cada item tem um nome, um preço e uma receita. Esta última estipula a quantidade de cada ingrediente (expressa nas unidades correspondentes) que entra na confecção do item. Alguns ingredientes são simples (ex: arroz ou açúcar) e outros são compostos (ex: molho bechamel ou chantilly). Sabe-se o preço de cada unidade de cada ingrediente. O preço de um menu é calculado com base na receita e na margem de lucro pretendida com esse menu. Pretende-se saber:

- R1 – o preço de custo de um dado menu
- R2 – a sobremesa mais cara feita com um dado ingrediente
- R3 – em quantos menus é incluído um dado item (por exemplo o prato “Bacalhau à Braz”)?
- R4 – o número de ingredientes da receita de um dado item (não é necessária a decomposição em ingredientes simples)
- R5 – o número de ingredientes simples da receita de um dado item (é necessária a decomposição)