

Disciplina	Cód. Disciplina	Data de Entrega do Projeto
Algoritmos e Lógica de Programação	IAL-002	17/11/2014
Nome do Trabalho: Estatísticas de Vendas		
Instruções: a) Leia e cumpra atentamente todos os itens dessas instruções; b) Leia, entenda e execute todos os requisitos listados na descrição deste Projeto-Programa; c) Na resolução deste Projeto-Programa é obrigatório usar a linguagem Pascal; d) Esta é uma atividade em grupo. Cada grupo deverá ter de 3 a 5 alunos. (Não serão aceitos trabalhos individuais ou em duplas) e) Este Projeto-Programa deve ser entregue no dia estipulado acima, no início da aula. f) Devem ser entregue um CD contendo o código fonte, o programa executável (dos dois programas) e os arquivos necessários para testes, se for o caso. O CD deve ser entregue acondicionado em embalagem de papelão, evitando caixinhas plásticas para evitar maiores volumes. A embalagem externa deve estar devidamente identificada com: <ul style="list-style-type: none">• Sigla da disciplina - IAL-002;• Nome do Trabalho - Estatísticas de Vendas;• Nomes completos de todos os integrantes do grupo - não é necessário colocar o número de matrícula; g) A não entrega deste projeto na data estipulada acarretará em nota N3 igual a zero, sem direito a qualquer reposição;		

Informações iniciais

Este projeto estará baseado na leitura de um arquivo texto contendo dados das vendas de produtos de uma loja. Este arquivo terá o nome **VENDAS.TXT** (obrigatório) e conterá os dados de venda organizados segundo o layout mostrado abaixo:

```
3001-3;12/03/2012;16;37.80;  
3519-0;12/03/2012;4;12.40;  
1958-2;13/03/2012;25;3.95;  
1210-2;13/03/2012;1;6.90;  
3519-0;13/03/2012;12;12.40;  
etc...
```

Como pode ser visto acima, cada linha do arquivo representa a venda de um item e contém quatro informações separadas pelo caracter "ponto e vírgula" (;). Essas informações são:

Posição	Informação	Formato	Observações
(1)	Código do Produto	4 dígitos numéricos + 1 dígito de verificação	O dígito de verificação é calculado utilizando-se o módulo 7 com pesos de 2 a 5. Veja abaixo o exemplo de cálculo
(2)	Data da Venda	Data no formato dd/mm/aaaa	
(3)	Quantidade Vendida	Número Inteiro	
(4)	Valor Unitário	Número Real	Usar o caracter ponto (.) como separador decimal

Exemplo de cálculo do DV do Código do Produto para o código 1958

Dígitos	1	9	5	8
Pesos	5	4	3	2
Cada dígito deve ser multiplicado pelo peso	5	36	15	16
Some todos os valores acima	5 + 36 + 15 + 16 = 72			
Calcule o resto da somatória por 7	Resto de 72 por 7 = 2			

Portanto o DV do código 1958 é 2 e o código completo ficará sendo 1958-2

Para dar continuidade à descrição deste projeto programa suponha que você possui um arquivo com todas as vendas efetuadas no mês de março/2012 e que este arquivo tenha 864 linhas. Considerando que o mês em questão tem 27 dias úteis (a loja não abre aos domingos), isso significa que a loja vende em média 32 itens por dia.

Este arquivo deverá ser lido pelo programa que vocês irão escrever e, em seguida, deve gravar um arquivo texto de saída conforme descrito abaixo. Para este arquivo deverá ser dado o nome **TOTVENDAS.TXT**

Formato do arquivo de Saída

Note que as três partes mostradas abaixo deverão estar gravadas no mesmo arquivo de saída

Parte 1 – Totais de Vendas por Dia: Deve ser gravado o seguinte

```
TOTAIS DE VENDAS POR DIA:
01/03/2012      1844,17
02/03/2012      2441,09
03/03/2012      4133,51
05/03/2012      1699,03
...
31/03/2012      3155,43
```

Parte 2 – Totais de Vendas por Produto: Deve ser gravado o seguinte

```
TOTAIS DE VENDAS POR PRODUTO:
1210-2          533,09
1215-5           57,31
1218-4         1045,09
...
8144-1          337,80
```

Parte 3 – Total e Estatísticas do Mês: Deve ser gravado o seguinte

```
TOTAL E ESTATÍSTICAS DO MÊS
Total de Vendas                    53457.38
Quantidade de produtos vendidos    3158
Média de Vendas por dia útil        1979.90
Média de Vendas por item            61.87
Média de Vendas por produto         16.93
```

Como obter cada valor acima? Pense um pouco e depois pense mais. Após tanto pensar e, caso ainda seja necessário, pergunte ao professor.

Como testar o seu programa

Na descrição acima, em determinado momento foi dito "...**suponha** que você possui um arquivo com todas as vendas efetuadas no mês de março/2012...".

Para aquele momento esta suposição estava boa. No entanto, agora é preciso dizer o seguinte: para testar o programa criado você **não tem esse arquivo**. E deverá ter. Basicamente, existem duas formas de criar um arquivo desses: digitá-lo à mão ou criar um programa capaz de gerá-lo.

Acredito que não seja boa idéia digitar um arquivo com mais de 800 linhas, contendo códigos, datas e números, portanto a primeira opção deve ser descartada desde já.

Como bons programadores vocês estão convidados a produzir um programa que gere o arquivo **VENDAS.TXT**. Este programa também faz parte do projeto e deve ser entregue no CD.

Requisitos do Programa Gerador (abreviado como PG, daqui em diante)

1. O PG deverá solicitar três dados de entrada ao usuário. São eles:
 - O Mês das vendas: deverá ser um número entre 1 e 12 (variável Mes);
 - O Ano das vendas: deverá ser um número maior que 2010 (variável Ano);
 - Quantas vendas são realizadas num dia: deverá ser um valor entre 25 e 50 (variável QtdeDia);
2. Descobrir quantos dias úteis (total de dias menos qtde. de domingos) existem no mês fornecido pelo usuário (variável DiasUteis). Para isso vocês deverão contar os domingos existentes. Descobrir se um determinado dia de um mês é um domingo não é nada complicado e uma dica para esse fim está anexada no final deste texto.
3. Com a variável DiasUteis carregada pode-se passar à geração do arquivo. O total de linhas que ele terá será DiasUteis * QtdeDia;
4. Para cada linha devem ser geradas quatro informações: Código do Produto, Data, Quantidade e Valor Unitário. Essas informações devem ser gravadas no arquivo separadas por "ponto e vírgula" (;). Com exceção da data, os outros três dados deverão ser gerados com a função **Random(ValorFinal)** que é um gerador de números pseudo aleatórios. Essa função gera um número inteiro entre 0 e ValorFinal-1. No começo do programa use a procedure **Randomize** para estabelecer a semente de geração dos números aleatórios, conforme foi explicado em aula.
5. Os códigos de produto devem ser um número entre 1100 e 1199. Para cada linha use a função Random para gerar o dígito verificador calculado pelo módulo 7 explicado anteriormente.
6. A data da venda não deve ser um número aleatório. Coloque QtdeDia linhas no arquivo para cada data que não seja domingo. O Mês e o Ano são fornecidos pelo usuário, assim, basta fazer um laço variando o dia de 01 a 30 (ou 31, ou 28, conforme o mês – ignorar bissextos, não é relevante neste caso).
7. A quantidade vendida de cada item deve ser um número inteiro entre 1 e 100 (usar Random também).
8. O valor unitário de venda deve ser um valor real entre 1.99 e 199.99 (usar Random também). A geração aleatória do valor de venda dos produtos acarreta em uma inconsistência. Essa inconsistência não afeta os testes que iremos realizar, mas alguém consegue identificar que inconsistência é essa? É possível propor uma solução para ela?

Função que informa qual é o dia da semana para uma data fornecida é a DayOfWeek

```
function DayOfWeek(DateTime: TDateTime): Integer;
```

mais informações em: <http://www.freepascal.org/docs-html/rtl/sysutils/dayofweek.html>

Esta função retorna um número inteiro, sendo 1 para Domingo, 2 para Segunda-feira, e assim por diante até 7 para o Sábado.

A função DayOfWeek deve receber uma data no formato TDateTime (que na verdade é um número Real) e para produzir uma data neste formato será necessário usar a função EncodeDate

```
function EncodeDate(Year: Word; Month: Word; Day: Word): TDateTime;
```

mais informações em: <http://www.freepascal.org/docs-html/rtl/sysutils/encodedate.html>

```
exemplo:
var
  aa, mm, dd: Integer;
  MinhaData: TDateTime;
begin
  aa := 2012;
  mm := 03;
  dd := 25;
  MinhaData := EncodeDate(aa, mm, dd);
  if MinhaData = 1 then
    writeln('Este dia é um Domingo')
  else
    writeln('Este dia não é um Domingo')
```