Mostrando as Datas Móveis

Mário Leite

...

Todos nós já observamos que a Quarta-feira de Cinzas sempre "cai" no dia imediatamente após à Terça-feira de Carnaval (!) Bem; esta observação, apesar de ser bastante óbvia, tem lá sua importância; pelo menos para começar esta publicação. Mas, talvez, o mais interessante seria fazer a seguinte pergunta: "Em que dia vai cair a Terça-feira de Carnaval de 2099?" Esta foi a pergunta que fiz a mim mesmo, depois de uma ressaca carnavalesca de Cuba Libre em 1985 quando, ainda, não existia os recursos práticos digitais para nos informar que será em 24/02/2099. Naquela época era muito difícil de saber, pois os calendários se restringiam aos dez próximos anos, e não a um século adiante; e além do mais, pairava sobre todos nós o pesadelo do bug do Século em 31/12/1999; os "velhinhos" do CoboL sabem muito bem do que estou falando, e para os mais jovens consultem o Google. E para mostrar a origem desta data festiva é importante retornar à Roma antiga, no ano de 325 D.C no Concílio I de Niceia, presidido pelo imperador romano Constantino I e sob o papado de São Silvestre I, que estabeleceu o seguinte: "O Domingo de Páscoa é o domingo subsequente à lua cheia que ocorre a 21 de Março, ou imediatamente após esta data; se a lua cheia ocorrer num domingo o Domingo de Páscoa será no domingo seguinte".

Esta resolução era meio complicada e também muito estranha; mas como era uma resolução imperial e abençoada pelo papa, os cientistas da época (notoriamente astrônomos e matemáticos) saíram à procura de fórmulas para tentar resolver o problema e apresentar um calendário de acordo com o que havia sido estabelecido; pois, brincar Carnaval fora de época naquela época dava fogueira, amigo! Um astrônomo napolitano e um jesuíta alemão converteram essa definição num algoritmo que foi demonstrado por Donald Knuth em seu livro "The Art of Computer Programming-Fundamental Algoritms" (vol. I; páginas 155-156). Outro procedimento computacional foi criado pelo grande matemático alemão Karl Friedrich Gauss; esse procedimento é demonstrado por William H. Glenn e Donovam Johnson no livro "Fun with Mathematics-Exploring Mathematics on your own" (John Murray publishers London pág. 22-27). Mas foi o genial Carl Friedrich Gauss que resumiu toda a complexidade dos cálculos numa simples tabela (vide tabela 1). Em função do Ano são usadas as constantes M ou N, e calculados os parâmetros A, B, C, D, E, F; onde mod é um operador que dá o resto da divisão.

```
A = Ano mod 4

B = Ano mod 7

C = Ano mod 19

D = (19*C + M) Mod 30

E = (2*A + 4*B + 6*D + N) mod 7

F = (22 + D + E)
```

De acordo com a **tabela 1** a análise a ser feita deve ser a seguinte:

Se **F** for maior que **31**, então o Domingo de Páscoa será no dia (**D** + **E** - **9**) de **Abril**; caso contrário, se **F** não for maior que **31**, o Domingo de Páscoa será no dia (**22** + **D** + **E**) de **Março**.

Em consequência desta análise podemos descobrir as outras datas móveis do seguinte modo:

```
    Terça-feira de Carnaval ==> 47 dias antes do Domingo de Páscoa
    Quarta-feira de Cinzas ==> 46 dias antes do Domingo de Páscoa
    Domingo de Ramos ==> 7 dias antes do Domingo de Páscoa
    Sexta-feira da Paixão ==> 2 dias antes do Domingo de Páscoa
    Ascensão do Senhor ==> 39 dias após o Domingo de Páscoa
    Corpus Christi ==> 60 dias após o Domingo de Páscoa
```

Como podemos notar, o algoritmo é de extrema simplicidade; mas tem um inconveniente: a sua validade expiraria no ano 2000; e como Gauss não informou a ninguém como chegou aos valores para **M** e **N**, a situação chegaria a um impasse. Entretanto, dando uma olhada mais detalhada na tabela proposta por ele, podemos estendê-la até o ano 2099, analisando os seguintes fatos:

- 1. O ano de 1600 foi bissexto, o mesmo aconteceu com o ano 2000; em contrapartida os anos 1700, 1800 e 1900 não foram bissextos;
- 2. A segunda coluna da tabela (que vai de 1582 a 1699) cobre dois séculos;
- 3. A última coluna proposta (1900 a 2000) não termina em 1999, conforme as anteriores.

Assim, se isso ocorre, o intervalo de anos para a última coluna poderá ser de 1900 a 2099, fazendo analogia com a segunda coluna, que vai de 1582 a 1699. Desta forma os valores **M=24** e **N=5** poderão também podem ser aplicados para o intervalo de 2001 a 2099; então, resolvi trabalhar com a tabela estendida até o ano de 2099 (vide **tabela2**). Deste modo, para o ano de 2099 temos **M=24** e **N=5**; e fazendo os cálculos, encontramos:

```
A = 2099 mod 4 = 3

B = 2099 mod 7 = 6

C = 2099 mod 19 = 9

D = (19*9 + 24) mod 30 = 15

E = (2*3 + 4*6 + 6*15 +5) mod 7 = 6

F = (22 + 15 + 6) = 43
```

Como **F** é maior que **31**, o Domingo de Páscoa de 2099 "cairá" no dia **(D+E-9)/04/Ano**, isto é, no dia **12/04/2099**; e as demais datas serão as seguintes:

```
    Terça-feira de Carnaval: (12/04/2099) - 47 ==> 24/02/2099
    Quarta-feira de Cinzas: (12/04/2099) - 46 ==> 25/02/2099
    Domingo de Ramos: (12/04/2099) - 7 ==> 05/04/2099
    Sexta-feira da Paixão: (12/04/2099) - 2 ==> 10/04/2099
    Ascensão do Senhor: (12/04/2099) + 39 ==> 21/05/2099
    Corpus Christi: (12/04/2099) + 60 ==> 11/06/2099
```

Como estamos no Windows[©] 10, hoje isto é muito fácil checar: basta acessar o calendário que está lá: **24/02/2099** - Terça-feira de Carnaval - que nos meus tempos de juventude era chamada de "Terça-feira Gorda", não sei porque!!!

Bem; E o que a programação tem a ver com isto!? Tudo, claro!

Retornando ao ano de 1985, quando me fiz a pergunta: *Em que dia vai cair a Terça-feira de Carnaval de 2099*", resolvi criar um programa para responder esta questão, já que conhecia seu algoritmo. Então, codifiquei o programa em Clipper Summer'85 e mais tarde o converti para o Delphi 7, cuja janela principal do programa pode ser vista na **figura 1**. Mas, ainda não satisfeito com a extensão da tabela original de Gauss até 2099, tentei estendê-la mais ainda: para até o final do terceiro milênio: ano 3000! Entretanto, nem tudo são flores na vida de um programador, pois, quando testei para o ano 2100 o programa falhou (comparando com o Calendário do Windows[©] 10); este calendário mostra Domingo de Páscoa no dia **28/03/2100**, e o meu programa mostra **27/03/2100**; isto é, meu programa dá UM DIA de atraso em todas as Datas Móveis. Quem será que está errado: EU ou o Windows[©] 10!?. Provavelmente eu; MAS, nunca é demais checar! Deste modo, resolvi colocar isto como um desafio aos programadores, já que meu programa acertou todas as datas até 2099. Seria um novo *bug* no ano 2100!?

Constante/Ano	1582 - 1699	1700 - 1799	1800 - 1899	1900 - 2000
М	22	23	24	24
N	2	3	4	5

Tabela 1 - Tabela Original de Gauss para Datas Móveis

Constante/Ano	1582 - 1699	1700 - 1799	1800 - 1899	1900 - 2000
M	22	23	24	24
N	2	3	4	5

Tabela 1 - Tabela Original de Gauss para Datas Móveis

Constante/Ano	1582 - 1699	1700 - 1799	1800 - 1899	1900 - 2099
М	22	23	24	24
N	2	3	4	5

Tabela 2 - Tabela estendida para Datas Móveis

Constante/Ano	1582 - 1699	1700 - 1799	1800 - 1899	1900 - 2099
M	22	23	24	24
N	2	3	4	5

Tabela 2 - Tabela estendida para Datas Móveis

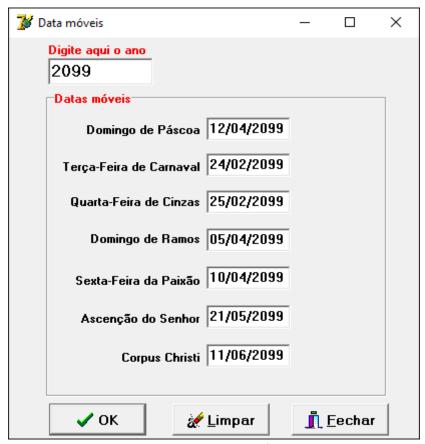


Figura 1- Obtendo as Datas Móveis de 2099



Figura 1- Obtendo as Datas Móveis de 2099

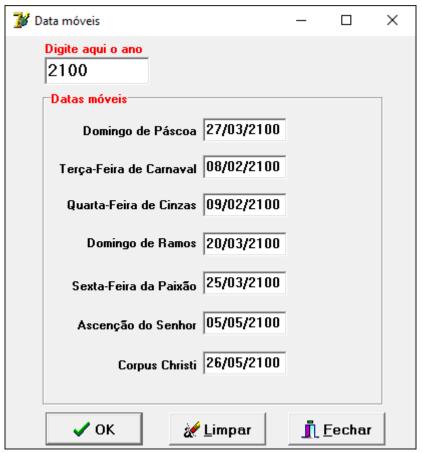


Figura 2- Tentativa de obter as Datas Móveis de 2100

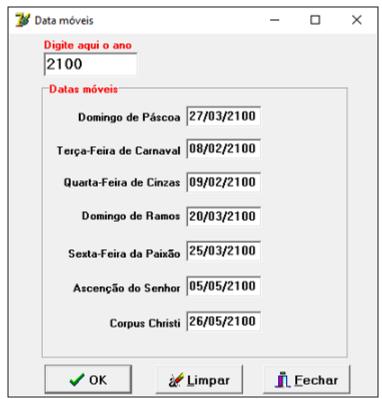


Figura 2- Tentativa de obter as Datas Móveis de 2100

```
//Programa "DatasMoveis"
//Mostra as Datas Móveis de um dado ano
//Implementado em Delphi
//Autor: Mário Leite
//marleite@gmail.com
unit UnDatasMoveis;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Buttons;
type
 TForm1 = class(TForm)
   EdtPascoa: TEdit;
   EdtCarnaval: TEdit;
   EdtRamos: TEdit;
   EdtPaixao: TEdit;
   EdtAscensao: TEdit;
   EdtCorpus: TEdit;
   Label1: TLabel;
   EdtAno: TEdit;
   GroupBox1: TGroupBox;
   Label2: TLabel;
   Label3: TLabel;
   Label4: TLabel;
   Label5: TLabel;
   Label6: TLabel;
   Label7: TLabel;
   BitExecutar: TBitBtn;
   Label8: TLabel;
   EdtCinzas: TEdit;
   BitLimpar: TBitBtn;
   BitBtn1: TBitBtn;
   procedure BitExecutarClick(Sender: TObject);
   procedure BitBtn1Click(Sender: TObject);
   procedure BitLimparClick(Sender: TObject);
 private
    {Private declarations}
 public
    {Public declarations}
 end;
war
 Form1: TForm1;
implementation
{$R *.dfm}
//-----
procedure TForm1.BitExecutarClick(Sender: TObject);
var DataPascoa, DataCarnaval, DataCinzas, DataRamos, DataPaixao,
   DataAscensao, DataCorpus: TDateTime;
   AnoStr, DataPascoaStr: string;
   AnoNum, A, B, C, D, E, F, DiaPascoaNum: integer;
```

```
begin
 //Captura o ano digitado pelo usuário
AnoStr := Trim(EdtAno.Text);
AnoNum := StrtoInt(AnoStr);
 //Calcula os parâmetros do algoritmo
A := AnoNum mod 4;
B := AnoNum mod 7;
C := AnoNum mod 19;
D := (19*C + 24) \mod 30;
E := (2*A + 4*B + 6*D + 5) \mod 7;
 F := (22 + D + E);
 //Determina a data do "Domingo de Páscoa"
if(F>31) then
   begin
     DiaPascoaNum := (D + E - 9);
     DataPascoaStr := Trim(IntToStr(DiaPascoaNum)) + '/04/' + AnoStr;
   end
 else begin
   DiaPascoaNum := (22 + D + E);
   DataPascoaStr := Trim(IntToStr(DiaPascoaNum)) + '/03/' + AnoStr;
 end;
 DataPascoa := StrToDate(DataPascoaStr);
 //Determina as datas móveis
DataCarnaval := DataPascoa - 47;
DataCinzas := DataPascoa - 46;
            := DataPascoa - 7;
DataRamos
DataPaixao := DataPascoa - 2;
DataAscensao := DataPascoa + 39;
DataCorpus := DataPascoa + 60;
 //Exibe as datas móveis
EdtPascoa.Text := DataPascoaStr;
EdtCarnaval.Text := DateToStr(DataCarnaval);
EdtCinzas.Text := DateToStr(DataCinzas);
                := DateToStr(DataRamos);
EdtRamos.Text
EdtPaixao.Text := DateToStr(DataPaixao);
EdtAscensao.Text := DateToStr(DataAscensao);
EdtCorpus.Text := DateToStr(DataCorpus);
end;
//-----
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
// Encerra a aplicação
Application. Terminate; //encerra a aplicação
end:
//-----
procedure TForm1.BitLimparClick(Sender: TObject);
//Limpas as caixas de texto e prepara para entrar com novo ano
var j: integer;
begin
 For j:=1 to ComponentCount -1 do
   If Components[j] Is TEdit then
      TEdit(Components[j]).Clear;
 EdtAno.SetFocus;
end;
end. //fim do programa
```