

Y2K e a Propriedade Intelectual / O *bug* do Milênio - Jan. / 1999

Direitos autorais de Clovis Silveira - reprodução autorizada contanto que citada a fonte.

ÍNDICE

- I. INTRODUÇÃO**
- 2. O PROBLEMA DO ANO 2000**
- 3. 2000 ANO BISSEXTO**
- 4. PROGRAMAS COM Y2K**
- 5. IMPLICAÇÕES TÉCNICAS**
- 6. APOCALIPSE ?**
- 7. CONCLUSÃO**

1 INTRODUÇÃO

O chamado "**Millenium Bug**" ou "**Bug do Milênio**" seria melhor denominado "**Bug do Ano 2000**" pois se refere basicamente ao mal funcionamento de computadores e/ou programas de computador que venham a interpretar equivocadamente **as datas do ano 2000**, quando codificadas na chamada "**convenção de dois dígitos**".

E é assim que vem sendo denominado este problema, de uma maneira geral, na bibliografia especializada : "**The year 2000 Computer Problem**" ou, abreviadamente, "**The Y2K Problem**".

O termo *bug* também merece algum comentário inicial, em particular para aqueles que não estejam familiarizados com os termos (e gírias) utilizados no campo das Ciências Computacionais.

"**Bug**", em inglês, é um *inseto*, um "*beetle*" e o adjetivo *buggy* significa "*infestado de insetos*" (embora "*buggy*" também possa ser *carro* -do Francês arcaico *bourger*- e até mesmo *carro de bebê*). Finalmente, "**bugs**", em inglês, é gíria para *louco* ou *insano*.

Mas "**bug**", na terminologia da Informática, é "**um erro ou mal funcionamento de um programa de computador ou de um sistema**".

Daí vêm também os termos "**debug**" e "**debugging**", que dizem respeito às **teorias e práticas computacionais de detectar, diagnosticar e corrigir erros de programas de computador** e que, a propósito, originaram o nosso neologismo, o verbo "**debugar**", dos mais utilizados pelos programadores, pois não há programa que não tenha tido **bugs**, e que não tenha sido **debugado**.

Finalmente, convém lembrar que o **milênio** que se aproxima não tem início no dia 1 de Janeiro do ano 2000, mas no dia **1 de Janeiro do ano 2001**.

Portanto vamos identificar a questão básica como sendo "**O Problema do Ano 2000**" ou, abreviadamente, o **problema Y2K**.

2. O PROBLEMA DO ANO 2000

Os computadores com “**programa armazenado em memória**” surgiram na “**década dos anos 60**”. Aliás, quando utilizamos a expressão “anos 60” nos referimos a um ano como, por exemplo, **1965** (e não **2065!**).

Em **1965** já estava instalado no Centro de Computação Eletrônica da Universidade de São Paulo um computador científico “IBM 1620”.

Naquela época, para se **armazenar um programa** na memória do computador, utilizavam-se, como meio de entrada, os cartões perfurados, codificados em 80 colunas (os denominados cartões “Hollerith”).

Cada **linha de código fonte de um programa** consumia em geral um cartão e, assim como o **programa**, também **os dados a serem processados eram “digitados”** nas máquinas perfuradoras de cartões, que tinham teclados igual ao de uma máquina de escrever, e semelhantes aos atuais teclados dos microcomputadores.

Um **computador** era **extremamente caro** (da ordem de milhões de dólares) e as **memórias** de armazenamento de capacidade ínfima, como, por exemplo, ridículos 20KB (20 mil bytes).

Contudo, o primeiro computador da revolucionária série /360 da IBM, instalado na IBM do Brasil em 1967 em São Paulo, tinha apenas 20 Kbytes de memória total! (isto não é suficiente, hoje, para se redigir uma boa carta de 5 páginas).

Os microcomputadores de hoje podem custar mil vezes menos (cerca de US\$ 1000) e podem ter memórias milhões de vezes mais capazes (2Gbytes = 2gigabytes = 2×10^9 bytes).

Portanto, qualquer economia em caracteres digitados e gravados em memória era extremamente importante. E é por isso que uma **data** como, por exemplo, **21 de Outubro de 1967**, não era representada por **21/10/1967** mas, por exemplo, por **21/10/67** ou, melhor ainda, simplesmente por **671021**, o que, **neste caso**, proporcionava uma **economia de cerca de 40% em memória** (economizados 4 em 10 caracteres ou bytes de memória).

Por isso tornou-se praxe a utilização da “**convenção de dois dígitos**” para codificar o ano de uma data, nos programas de computador em geral. Essa convenção também foi aplicada aos programas gravados em *chips* e, hoje, estão nos “calendários” internos (os *clocks*) dos microcomputadores.

Assim, quando chegarmos ao dia **1º de Janeiro de 2000**, utilizando a mesma convenção (nós estaríamos então escrevendo **01/01/00** e não 1/1/2000)

ocorrerá, por exemplo, que os computadores que tratam os anos das datas com apenas dois dígitos, **interpretarão 1/1/00 como sendo 1/1/1900 ou seja, 1 de Janeiro de 1900**, cometendo, então, um erro de cem anos para menor.

Uma conseqüência imediata, portanto, é que todas as **datas posteriores a 31/12/1999** serão interpretadas, **pelos computadores e/ou programas que apresentam o “bug Y2K”** (seja no *hardware* ou no *software*) **com um erro de menos 100 anos**.

Ou seja, quando a data **31/12/1999 mudar para 01/01/00** os computadores "imaginarão" que se encontram em **01/01/1900**.

Assim, o cálculo da **idade** de uma pessoa nascida neste século, por exemplo, resultará em **número negativo**: alguém nascido em 1967, que terá no ano 2000, efetivamente **33** anos, terá sua idade calculada, pelos computadores ou *softwares* com *bug Y2K*, como sendo **-67 anos**.

Os cálculos **de tempos decorridos**, que dependam de **eventos** que tenham datas que utilizaram a convenção dos dois dígitos e/ou que estejam sendo processadas por computadores que não tenham a capacidade de lidar senão com datas que utilizam a convenção dos dois dígitos, em seu *hardware*, **resultarão grosseiramente errados**; e quanto **mais próximo o evento do ano 2000 maior será o erro em anos (valor absoluto)**.

Assim, qualquer **cálculo pro-rata** como os de juros, os de indenizações, e outros, resultarão em absurdos valores e em lançamentos absurdos em contas bancárias, em outros milhões de absurdos, e assim por diante.

Então, essa é a essência **do problema Y2K**

3. ANO 2000 BISSEXTO

Outra questão que tem sido recentemente colocada como sendo mais um “Bug do Milênio” é o chamado **problema do ano 2000 bissexto** (Y2K LEAP YEAR PROBLEM).

A maioria das pessoas sabe que “**ano bissexto**” é aquele em que é **acrescido um dia no mês de Fevereiro**, quando então o ano passa a ter o dia **29 de Fevereiro**.

A maioria das pessoas sabe que os anos bissextos ocorrem em **anos múltiplos de 4**. Mas há, também, quem pense que a cada 100 anos deveria ser acrescido, ainda outro dia em Fevereiro, para fazer uma correção centenária..

Estes últimos têm chegado à conclusão que o **ano 2000**, por ser **divisível por quatro** e também ser **múltiplo de 100**, deveria ser um **ano bissexto especial**, que teria o **dia 30 de Fevereiro!**.

E então, dizem, além do **bug do dia 1/1/2000**, acima explicado, os computadores e programas apresentariam ainda **outro grande bug em 30 de Fevereiro**, por que o dia 30 de Fevereiro, não estando previsto nos calendários automáticos dos computadores, implicaria em novos erros de cálculos de datas e de tempo decorrido entre datas.

Este “bug”, na realidade, não existe. Trata-se de um erro daqueles que não conhecem bem a estrutura do calendário Gregoriano e o que são os **anos bissextos**.

Bissexto era o **nome do dia** que adicionado **ao mês de Fevereiro** (para correção do calendário Juliano), termo que foi extendido para designar o **ano** que tem o “dia bissexto”.

O **dia bissexto** era o dia que se adicionava após o sexto dia anterior ao **“calenda” de Março** (que era o nome primeiro dia do mês, dia em que os devedores deviam pagar aos credores), para efetuar a correção do “calendário Juliano”. O ano Juliano tinha **365 dias e seis horas** ou seja, **365 e _ de dia**. Daí a correção de **um dia a cada 4 anos**.

Na época do **Papa Gregório** observou-se que a correção de um dia a cada quatro anos, realizada para ajustar o calendário civil ao ano solar, não estava sendo suficiente, pois já era constatado um **erro efetivo de dez dias**, motivo pelo qual foi proposta a **reforma do calendário** cristão.

O **calendário Gregoriano** foi instituído no Mundo Cristão em 1581 e passou a ter vigência em 4/10/1582, tendo sido, de pronto, suprimidos os 10 dias em excesso.

Assim, **o dia seguinte ao 4/10/1582 foi o 15/10/1582!** o que causou grande confusão e discussões, na época, principalmente entre credores e devedores.

A partir de 1582 o Calendário Cristão passou a ser corrigido com a inclusão do **29 de Fevereiro** nos chamados **anos bissextos** (em inglês "leap years"), que ficaram definidos com as seguintes regras:

- a) **todo ano não centenário múltiplo de 4 é bissexto;**
- b) **todo ano centenário divisível por 400 é bissexto.**

Assim, o ano **1600** foi bissexto;
1700 ,1800 e 1900 não foram;
2000 será bissexto;
2100, 2200 e 2300 não serão bissextos.
2400 será.

Em resumo, o ano **2000**, como qualquer outro **ano bissexto**, terá o dia 29 de Fevereiro. No calendário Gregoriano nunca existirá um dia 30 de Fevereiro. O dia seguinte ao 29/2/200 será o 1/3/2000.

Não só como curiosidade - mas talvez seja oportuno que os programadores e fabricantes de chips saibam desde já : o próximo ajuste "especial" do Calendário Gregoriano (com a supressão de um dia do ano) ocorrerá no **ano 4915**. Esse ajuste deve ocorrer a cada 3333 anos e 1/3, após o início do calendário Gregoriano, ou seja, após 1582).

Será o **BUG do Quinto Milênio?**

4. PROGRAMAS COM Y2K

Imagine-se um **programa**, ou uma **rotina de um programa**, que pode inclusive estar embutida num **chip** de um dispositivo qualquer, dos mais simples, caseiros, ou um dispositivo de controle mais importante, de uma usina de geração energia, de um de sistema de controle de tráfego aéreo, etc.

A rotina foi escrita para realizar alguma coisa, **disparar algum processo, em intervalos de tempo regulares**. Foi escrita numa linguagem de alto nível qualquer, mais ou menos assim:

1. pegue "data/hora" no **timer** e armazene em "antes"
2. enquanto a condição for verdadeira, continue em ação
3. pegue "data/hora" no timer e armazene em "agora"
4. se "agora" - "antes" > 10 execute "processo"
5. faça "antes" = "agora" e vá para linha 1

A rotina acima equivale ao seguinte: **"fique verificando continuamente se já se passaram dez segundos desde que o "processo" foi executado pela última vez. Se for o caso, execute-o novamente e memorize quando o fizer, para a nova verificação"**.

O "**processo**" pode ser qualquer ação que um computador necessite disparar regularmente, como por exemplo, um controle de **semáforo**, para verificar se é momento de apagar/acender novamente, uma **usina nuclear** para ver se é o momento de verificar a temperatura do núcleo central, etc

E o que isto tem a ver com o bug do ano 2000?

Enquanto os momentos **agora** e **antes** estiverem neste século tudo estará funcionando bem, mas logo que **agora** se tornar um momento após o início de 1/1/2000 e **antes** continuar em 1999, o cálculo das diferenças **agora - antes** ficará **negativo** e não se executará nunca mais o **processo**.

Portanto, toda as rotinas desse tipo, que tratam de **tempo relativo**, como a exposta, **deixarão de funcionar**, na **transição para o ano 2000**, com consequências imprevisíveis.

À primeira vista, poderia parecer um programa inofensivo. Mas a realidade é que muitos programas têm esse tipo de problema de **transição para o ano 2000**.

Se **chips** contiverem tais tipos de bugs, necessitarão ser fisicamente **substituídos**. **Programas** necessitarão ser **reparados**.

Há equipamentos, como por exemplo os sistemas de calefação de edifícios e os sistemas de controle de tráfego, que possuem controles computadorizados que dependem de **outro tipo de programa inofensivo**, para o cálculo do **dia da semana**, em função do que realizam **algum processo especial**. Estes equipamentos também vão falhar, mas talvez com consequências não tão desastrosas.

Isto porque quando o ano 2000 é interpretado como sendo 1900, os **dias da semana** não estarão corretos, fazendo com que rotinas de dias de semana sejam executadas em fins de semana e vice versa, por exemplo. É interessante notar que, para esse caso, se mudarmos a data interna do computador de 2000 para 1972, esse tipo de erro ficará (precarosamente) corrigido.

Já **outros tipos de programas** utilizam extensivamente as **datas**, propriamente ditas, como é o caso dos **sistemas financeiros** que, com certeza, têm que ser corrigidos em tempo, para que possam passar pela transição do ano 2000, sem produzir verdadeiros desastres no mercado.

Finalmente, é importante observar que, embora grande quantidade de sistemas e de computadores venha a apresentar defeitos de funcionamento, **à meia noite do dia 31 de Dezembro de 1999**, devido ao sistema de tratamento de datas de dois dígitos, como já foi exaustivamente comentado, muitos outros computadores o farão em outros horários, porque utilizam sistema de datas totalmente diferente, como é o caso do Sistema de Hora Universal (GMT Greenwich Mean Time) **internamente**.

Estes **falharão em horários locais diferentes de meia noite**, dependendo da longitude do local. Por exemplo, no Japão, às 19:00, e no Brasil às 7 da manhã.

5. IMPLICAÇÕES TÉCNICAS

Além das questões decorrentes da transição para o ano 2000, acima indicadas, que implicam na substituição de **chips** com programas gravados ou na correção de programas em sistemas de computador, existe a necessidade de se realizar **modificações estruturais em bancos de dados**, em **protocolos de comunicação**, em **interfaces de usuários**. etc. sempre que a existência de datas na convenção de dois dígitos esteja presente e produza resultados errados.

Não basta corrigir os programas que fazem acesso a um banco de dados, se também não forem corrigidos os próprios conteúdos dos bancos de dados. Porque um programa corrigido para lidar com datas de quatro dígitos cometerá erros ao buscar datas de dois dígitos, num banco de dados que ainda não tenha sido atualizado.

Por outro lado, não será possível operar bancos de dados corrigidos para quatro dígitos com programas que ainda não o foram.

Parece então ser uma tarefa impossível, pois, para tudo funcionar, deveria ser tudo corrigido de uma só vez. Aliás, como um programa preparado para a correção de um banco de dados poderá saber e decidir, processando enormes quantidades de correções em escala, se um determinado ano de dois dígitos, hoje, como por exemplo, **20**, significa **1920** ou **2020**?

Provavelmente algumas empresas resolverão utilizar regras práticas, e nem sempre precisas, para se aproximar de uma solução, como, por exemplo adotar que, se o ano é um número pequeno de dois dígitos (digamos menor do que 50), deve-se somar 2000, e se for um número maior do que 50, deve-se somar 1900, para torná-los de quatro dígitos.

As buscas em bancos de dados, que se referem a seleções de registros que envolvem datas, nunca serão confiáveis. Como por exemplo, uma busca para obter como resposta uma listagem de patentes que têm sua vigência entre duas datas.

Outra questão complicada reside nos **protocolos de comunicação**, necessários para a conexão de computadores em redes. Os protocolos de comunicação carregam datas no formato do dois dígitos, precisarão ser redesenhados. Este também é um problema que não é de simples solução pois o redesenho dos protocolos requer seja corrigidos adequadamente os programas que utilizam o protocolo.

Para finalizar esses comentários técnicos, é interessante observar que sistemas modernos, como o Windows e o Unix, e alguns sistemas de banco de dados, utilizam um **método peculiar de armazenamento de datas**.

Estes sistemas, ao armazenarem uma data/hora, o fazem transformando-a em **segundos de tempo decorridos** desde uma **data base**, em geral **1970**. Assim, por exemplo, 9:36:09 PM do dia 21/11/1998 seria armazenado como o número inteiro 911.712.969 (segundos, desde zero hora de 1 de Janeiro de 1970).

Esse sistema não tem os problemas acima citados de transição do ano 2000, não tem problemas de dias da semana e nem os de anos bissextos, pois o cálculo de diferenças de datas sempre será correto, as datas do ano 2000 em diante etc.

Mas apresenta dois **outros bugs peculiares**.

O primeiro **bug** é que tais sistemas não podem armazenar datas superiores a 18 de Janeiro de 2038! pois esta data corresponde ao número 2 bilhões, que é o maior número inteiro que pode ser armazenado num computador de 32 bits.

O outro **bug** é que neste formato não é possível armazenar datas anteriores a 1970, pois muitos sistemas não aceitam armazenar números inteiros negativos; mas, mesmo que permitissem, estariam limitados ao ano **1902**, o que corresponde a -2bilhões de segundos com relação à base 1970.

Isto significa que mesmo os mais modernos sistemas Windows poderão falhar ao tentar tratar o ano **1900**, o que ocorrerá quando uma interface de usuário de um programa converter a data **1/1/00 em 1/1/1900**, o que o software tentará converter em segundos com relação a 1970, e será impossível. O software então ou enviará uma mensagem ao usuário ou simplesmente "dará pau".

6. APOCALIPSE?

Diversos sinais do bug do ano 2000 já vêm ocorrendo; não é preciso aguardar o dia da transição:

- Uma firma americana processou seu fornecedor de software em US\$250.000 porque o sistema desenvolvido não será capaz de identificar datas de expiração posteriores a 1 de Janeiro de 2000 em cartões de crédito.
- A Boeing descobriu que os **checks** de partida das aeronaves não funcionarão adequadamente na data de transição, se não forem corrigidos em tempo.
- Certos dispositivos para tratamento radioativo, se não substituídos antes de 1/1/2000, emitirão dosagens incorretas (baixas) de radiação nos pacientes.
- No dia 1 de Janeiro de 1999 os computadores do aeroporto de Arlanda, na Suécia, não conseguiram emitir passaportes temporários, devido ao problema Y2K.
- no dia 1 de Janeiro de 1999 os metrônomos dos taxis de Singapura deixaram de funcionar devido ao problema Y2K.
- Muitos computadores pessoais não funcionarão corretamente na transição 1999 para 2000, e reverterão o *clock interno* para o **ano 1970 ou 1980**.

Especialistas norte-americanos nas questões Y2K têm procurado analisar os possíveis cenários que deverão ocorrer durante o ano de 1999, na transição para o ano 2000 e daí para a frente.

O governo norte americano tem trabalhado com cenários em que se prevêem falhas ou paradas em sistemas pesados tais como os sistemas de geração de energia, os sistemas de telecomunicações, os sistemas de tráfego aéreo e os sistemas financeiros.

Esse trabalho já vem sendo realizado há vários anos, e a agências governamentais tem recebido instruções de como proceder para corrigir os programas de computador.

Contudo as empresas privadas estão numa situação delicada.

De uma maneira geral as empresas não estão liberando muita informação sobre sua situação com relação ao problema Y2K, pois se admitirem que têm um problema muito grande, suas ações perderão valor e se afirmarem que têm seus problemas Y2K já resolvidos, poderão ser judicialmente processadas, no futuro.

Segundo alguns autores, embora, de uma maneira geral, os **projetos Y2K**, ou seja, **os projetos internos das empresas para a correção de seus programas**

de computador e preparação para a transição do ano 2000, já deveriam ter terminado no final de 1998, e o ano de 1999 deveria ser inteiramente dedicado aos testes dos sistemas, o que deverá ocorrer a esse respeito é o seguinte:

- Muitas empresas de porte médio e pequeno não conseguirão ficar preparadas, uma vez que nem começaram seus projetos, e cerca de 50% tem respondido às enquetes que não pretendem fazer nada até lá.

- Muitos departamentos federais e estaduais do governo norte-americano também não estarão preparados, pelo fato de possuírem sistemas muito grandes e estarem com seus projetos internos atrasados.

- Muitas ações na Justiça deverão ocorrer. As despesas com processos deverão se tornar muito importantes à medida em que os problemas forem aparecendo em 1999 e de 2000 em diante.

- Muitas pessoas serão afetadas pelo problema Y2K, sem chegarem a entender o que aconteceu, e nem saber o que fazer.

Transcrevo, a seguir as **Previsões para os próximos 18 meses**, divulgadas pela Y2KNewsWire, na Internet:

1. Haverá pelo menos uma colisão de aviões ou acidente na aterrissagem ou decolagem, devido ao problema Y2K.

2. Pelo menos um caso de operação incorreta de sistema de tratamento de água ocorrerá causando doenças, devido ao problema Y2K.

3. Haverá falta de alimentos em muitas regiões, que serão atendidas por forças do Exército ou Cruz Vermelha.

4. Haverá declaração de Emergência Nacional, pela Presidência da República.

5. Haverá corridas a Bancos.

6. Ocorrerá pelo menos uma ação terrorista fatal no território americano ou fora, uma vez que inimigos dos americanos vão tirar proveito do problema Y2K.

7. O Departamento de Defesa sofrerá falhas de comunicações.

8. Pelo menos uma usina nuclear parará de funcionar em função de problema Y2K relacionado à segurança.

9. Os preços da gasolina vão subir devido a interrupções no fornecimento de óleo, causadas pelo problema Y2K.

10. Haverá muitas mortes em função de falhas em equipamentos médicos.

11. A falta de energia elétrica afetará pelo menos dez cidades norte-americanas.

12. Em Março de 1999 muitas agências federais norte-americanas anunciarão que já estão preparadas para a transição 1999/2000. No ano seguinte, metade delas falhará.

13. Ocorrerá um pico de corrida de pânico às lojas, em Novembro de 1999, especialmente comprar enlatados e geradores de energia.

14. Serão impostos limites para retiradas de dinheiro e para transações.

15. As vendas de armas aumentarão enormemente no verão e no outono de 1999, sobrecarregando os sistemas de consulta do FBI.

16. Muitos automóveis deixarão de funcionar corretamente.

17. A taxa de desemprego deverá superar 12% antes do final do ano 2000, em função de falências de empresas causadas pelo problema Y2K.

18. Ao final do ano 2000 a economia norte-americana estará em recessão (no mínimo).

19. O índice Dow cairá abaixo de 6000 devido, em parte, ao medo do problema Y2K e, em parte, às falências de empresas em decorrência do problema Y2K.

20. As ferrovias ficarão paralisadas pelo menos três vezes em certas áreas, causando demoras e parando indústrias.

21. Alguns segmentos da Internet não sofrerão mas as falhas de comunicações derrubarão a Internet em muitas áreas.

22. Pelo menos uma rebelião ocorrerá em presídio, causando muitas mortes.

23. Numa grande cidade, pelo menos uma vez, a paralisação dos semáforos causará congestionamento total.

24. Antes do verão do ano 2000 haverá forças do Exército policiando as ruas de pelo menos uma cidade norte americana.

7. CONCLUSÃO

Parece que o problema Y2K é um problema real.

Há centenas de empresas trabalhando nessa questão, com equipes que chegam a milhares de pessoas e já foram gastos milhões de dólares. Mas há muita gente que ainda não acredita que há um problema real...

O problema é real e oferece **maior risco** para as **empresas multinacionais, para os países que dependem mais de computadores e que se iniciaram mais cedo....**

Também têm grande risco os países que importam commodities e os que depende de receitas de exportação.

A única **conclusão** plausível a que se pode aqui chegar, é que **a questão Y2K necessita ser conhecida, discutida, e as empresas e as pessoas precisam estar suficientemente preparadas para enfrentar as eventuais dificuldades que provavelmente ocorrerão na transição para o ano 2000.**