

**ELABORAÇÃO DE UM CRONÔMETRO, EM TEMPO
REAL, PARA UTILIZAÇÃO EM APLICAÇÕES
HIPERMÍDIA IMPLEMENTADAS COM *TOOLBOOK*¹**

*DESIGN OF A CHRONOMETER, IN REAL TIME, TO USE IN
HYPERMIDIAS CREATED WITH TOOLBOOK*

**Lucas Mariano Schneider², Alberto Pessoa da Costa³
e Gilberto Orengo de Oliveira⁴**

RESUMO

Neste trabalho, é descrita a implementação de um cronômetro digital, para uso em um aplicativo hipermissão, construído na plataforma do *software* ToolBook. A principal contribuição está na simplicidade do *script* desenvolvido, o qual permitirá a utilização do cronômetro para contagem de intervalos de tempo, em tempo real, sendo que, com pequenas adaptações, é possível a implementação em outras linguagens de programação.

Palavras-chave: ToolBook, cronômetro, *script*.

ABSTRACT

In this work is described the implementation of a digital chronometer to use in hypermidias, constructed with ToolBook software. The main contribution is in the simplicity of script developed, which will allow its use for counting of time intervals, in real time, and with small adaptations is possible the implementation in other programming languages.

Keywords: ToolBook, chronometer, *script*.

¹ Trabalho de Iniciação Científica – PROBIC.

² Acadêmico do Curso de Física Médica – UNIFRA.

³ Co-orientador - UNIFRA.

⁴ Orientador – UNIFRA.

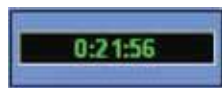
INTRODUÇÃO

O *ToolBook* [ASYMETRIX 1999], *software* utilizado para criação de aplicações hipermídia, possui uma programação direcionada para objetos. Nele é possível criar aplicações que são divididas em páginas, contendo objetos, tais como botões, textos, animações, vídeos e sons, entre outros. Este *software* possui linhas de programação, *scripts*, nos quais se programam ações e, ou modificações sobre o próprio aplicativo e sobre os objetos contidos, em uma linguagem de programação chamada de “*open script*” [CORTÊS, 1997].

Neste trabalho, é descrito o desenvolvimento de um cronômetro, um objeto utilizado em *ToolBook*, de programação simples e clara, que, facilmente, pode ser adaptada a qualquer aplicação desenvolvida em *ToolBook* e, com algumas modificações, a outras linguagens de programação.

O CRONÔMETRO

A criação deste cronômetro é constituída por dois aspectos: a construção de um campo de texto, no qual serão exibidos os intervalos de tempos (Figura 1a), e o script (Figura 1b), responsável por transformá-lo em um contador digital.



(a)

```

Script for Field "TEMPOHAS" of Page "BANCO DE DADOS"
File Edit Format View Window Help
NOTIFYBEFORE_IDLE -- instruções que serão repetidas continuamente
SYSTEM T
SYSTEM LOOP IN
IF ECOSYSTEME THEN
    TH = TH + 1 -- é armazenado em "TH" o tempo total em segundos
END
T-SYSTEME -----TRANSFORMANDO EM H:M-----
s = TH -- segundos
m = 0 -- minutos
h = 0 -- horas

do
    if (s>=60) then -- se o n° de segundos for >= 60 ...
        s = s - 60 -- ... a cada 60 segundo retirados...
        m = m + 1 -- ... é sonado 1 minuto
    end if
    if (m>=60) then -- se o n° de minutos for >= 60 ...
        m = m - 60 -- ... a cada 60 minutos retirados
        h = h + 1 -- ... é sonado 1 hora
    end if
until (s<0) -- só termina os ciclos se o n° de segundos for < 60
TEXT OF SELF = h & ":" & m & ":" & s -- h:m:s
    
```

(b)

Figura 1 - (a) Campo de texto e (b) seu *script*, componentes responsáveis pelo funcionamento do contador de tempo.

A estratégia adotada para a programação do cronômetro, que está explicitada nas seis primeiras linhas do *script*, pode ser descrita da

seguinte forma: inicialmente, ele captura o horário do relógio interno do sistema do computador, no formato hora:minuto:segundo. A seguir, repete-se o mesmo procedimento, em um intervalo de milésimos de segundo, e compara-se este valor com o obtido anteriormente. Se forem iguais, nada é executado, mas se forem diferentes é somada uma unidade ao valor de uma variável, previamente definida. Dessa forma, a variável funciona como um contador de segundos, pois cada vez que o tempo do sistema mudar (valores anterior e posterior diferentes), é porque transcorreu um segundo. Assim, a essencial função de um cronômetro, que é registrar o tempo real transcorrido, é satisfeita.

Cotidianamente, porém, os intervalos de tempo não são fornecidos somente em segundos, mas sim, no formato hora, minuto e segundo. Então, foram desenvolvidas as últimas quatorze linhas do *script*, que constituem um algoritmo para transformação de segundos no formato H:M:S.

A EVOLUÇÃO DA PROGRAMAÇÃO SCRIPT

O SCRIPT INICIAL

O *script* criado inicialmente é apresentado a seguir. Ainda não contém a transformação para o formato H:M:S e corresponde a um campo de texto com nome de “TEMPOHMS”.

```
NotifyBefore Idle      -- linha 1
System t              -- linha 2
System Long TM       -- linha 3
  IF (t<>Systime) THEN -- linha 4
TM = TM + 1           -- linha 5
  END                  -- linha 6
t = Systime           -- linha 7
Text of self = TM     -- linha 8
End                   -- linha 9
```

Na linguagem do *ToolBook*, a *Open Script*, o comando “**NotifyBefore Idle**” indica o início de um bloco de instruções que serão repetidas continuamente. Esse bloco é finalizado pela instrução “**End**”, na linha 9. O comando “**System**” é utilizado para declaração de variáveis globais, que podem ser utilizadas em vários *scripts*, e o “**Long**”, na linha 3, é um parâmetro que indica que a variável é do tipo inteira longo (32-bit). O “**IF**”, na linha 4, é o comando usado para criar uma condição, na qual o

termo “**THEN**” delimita o argumento. Somente serão executadas as linhas compreendidas entre o “**IF**” e o “**END**”, na linha 6, se a expressão lógica do argumento for verdadeira. O “**Systime**”, linhas 4 e 7, é uma função que captura o horário instantâneo do sistema do computador.

No processo, são declaradas duas variáveis, “**t**” e “**TM**”, em que “**TM**” armazenará valores inteiros e “**t**” será do tipo do primeiro valor que lhe for atribuído. No primeiro ciclo, “**t**” não possuirá nenhum valor, então, será diferente de qualquer valor que a ela for comparado. Ao chegar à linha 4, será feita a comparação entre o valor de “**t**” e o valor de “**Systime**”, horário do sistema, sendo eles diferentes, a variável “**TM**” será incrementada de um. Ao sair da condição “**IF**”, na linha 6, é atribuído à variável “**t**” o valor referente ao “**Systime**”. Então, no próximo ciclo, a variável “**TM**” somente será incrementada se tiver passado um segundo, ou seja, se “**t <> Systime**” (se “**t**” for diferente de “**Systime**”). Com isso, “**TM**” torna-se um contador de segundos. A linha 8 torna o texto do campo igual ao valor de “**TM**”, exibindo o tempo. Os caracteres “**--**” que antecedem as linhas de instrução indicam comentário, isto é, não serão executadas.

CRIANDO AÇÕES SOBRE O CRONÔMETRO

Até o momento, somente foi exibida a possibilidade de criação e funcionamento de um cronômetro, mas como manipulá-lo? Com esse algoritmo, é muito simples criar botões ou ações que possam iniciar ou parar a contagem. A seguir, serão apresentados alguns scripts para serem utilizados em botões, mas que também podem ser adaptados e utilizados em qualquer ação possível no *ToolBook*.

Para criar um botão que inicie a contagem o *script* do cronômetro foi modificado, como mostram as seguintes linhas:

```

NotifyBefore Idle          -- linha 1
System t , start          -- linha 2
System Long TM            -- linha 3
  IF (start = “ok”) THEN -- linha 4
If (t<>Systime) Then      -- linha 5
TM = TM + 1                -- linha 6
  End                       -- linha 7
t = Systime                -- linha 8
END                        -- linha 9
text of self = TM         -- linha 8
End                       -- linha 10

```

Neste script, foi adicionada uma condição para que o cronômetro comece a contagem. Agora, para iniciar, é necessário que a nova variável “**start**” seja igual a “**ok**”. Nessas condições, um botão que inicie a contagem deve possuir o seguinte *script*:

```
To handle ButtonClick           -- linha 1
System start                   -- linha 2
  start = “ok”                   -- linha 3
End                             -- linha 4
```

Nesse, o comando “**To Handle ButtonClick**” corresponde à ação de clicar sobre o botão. Então, quando o botão for clicado, “**start**” será igual a “**ok**” e o cronômetro iniciará a contagem. Para ter início outra ação, essa deve conter, em seu script, as linhas 2 e 3.

Um botão, que pare ou reinicie a contagem, deve possuir o seguinte script:

```
To handle ButtonClick           -- linha 1
System start , TM               -- linha 2
IF (start = “ok”) THEN         -- linha 3
  Start = “pause”               -- linha 4
ELSE                             -- linha 5
  TM = 0                         -- linha 6
Text of field “TEMPOHMS” = “0” -- linha 7
END                             -- linha 8
End                             -- linha 9
```

Neste botão, com um clique é acionada a parada da contagem. Se o botão for novamente clicado, o cronômetro será reiniciado e a contagem começará a partir do zero. Na linha 3, é verificado se o cronômetro está ativo, ou seja, se “**start**” é igual a “**ok**”; é executada a linha 4 que atribui o valor de “**pause**” à variável “**start**” (que é diferente de “**ok**”), parando a contagem. Caso contrário, o comando “**ELSE**” (interno à estrutura “**IF**”) indicará que o contador já está parado e executará as linhas 6 e 7, zerando o contador “**TM**” e o campo de texto do cronômetro.

A IMPLEMENTAÇÃO DA CONTAGEM REGRESSIVA

O cronômetro faz somente contagem de tempo progressiva, mas dependendo do aplicativo, pode ser necessário que a contagem de tempo

seja regressiva. Um exemplo é o de uma aplicação que executa um teste de múltipla escolha, no qual o usuário dispõe de um tempo determinado para concluir sua resposta. O processo, para criação da contagem regressiva, é muito semelhante ao desenvolvido anteriormente, modificando-se apenas alguns detalhes no script do campo de texto do cronômetro, como mostra o novo script:

```

NotifyBefore Idle           -- linha 1
System t                    -- linha 2
System Long TM              -- linha 3
If (TM <> 0) Then           -- linha 4
  IF (t<>Systime) THEN      -- linha 5
    TM = TM - 1             -- linha 6
  END                        -- linha 7
End                           -- linha 8
t = Systime                  -- linha 9
text of self = TM           -- linha 10
End                           -- linha 11

```

Nesta programação, a variável “**TM**”, se diferente de zero, está sendo diminuída de uma unidade a cada segundo. No entanto, deve-se estipular um tempo inicial por alguma ação. No caso do exemplo citado anteriormente (teste de múltipla escolha), o usuário terá cinco minutos para concluir o teste após entrar na página da questão 1. Portanto, o *script* da página deve conter o seguinte:

```

To handle EnterPage         -- linha 1
System Long TM              -- linha 2
TM = 300                    -- linha 3
End                           -- linha 4

```

O comando da linha 1 refere-se à ação de entrar na página. Quando essa ação é executada, a variável “**TM**” tem o valor de 300, que é o valor de segundos correspondentes aos cinco minutos.

O SCRIPT FINAL: em Horas, Minutos e Segundos

A transformação do formato do tempo é necessária para melhor visualização. Não estamos acostumados a olhar para o relógio e dizer que se passaram 41400 segundos, mas sim, 11 horas e 30 minutos. Então, o

intervalo de tempo apresentado, no cronômetro, deve ser transformado para esse formato. O *script*, para o campo de texto do cronômetro, entre as linhas 8 e 23, são apresentadas a seguir:

```
NotifyBefore Idle -- 1
System t -- 2
System Long TM -- 3
IF (t<>Systime) THEN -- 4
TM = TM + 1 -- 5
END -- 6
t = Systime -- 7
-----TRANSFORMAÇÃO----- -- 8
Local s , m , h -- 9
s = TM -- s é o tempo em segundos --10
m = 0 -- m é o tempo em minutos --11
h = 0 -- h é o tempo em horas --12
do -- comando de inicio de ciclos --13
if (s>=60) then -- se s >= 60 ... --14
s = s - 60 -- ...diminui-se 60 segundos e... --15
m = m + 1 -- ... soma-se 1 minuto. --16
end if --17
if (m>=60) then -- se s >= 60 --18
m = m - 1 -- ...diminui-se 60 minutos e... --19
h = h + 1 -- ... soma-se 1 hora. --20
end if --21
until(s<60) -- se s<60 os ciclos são encerrados --22
text of self = h &":":& m &":":& s --h:m:s --23
End --24
```

Esta transformação também pode ser acoplada nos *scripts* citados anteriormente, na contagem regressiva e nas ações sobre o cronômetro.

CONCLUSÃO

Se forem utilizados os *scripts* descritos neste trabalho, é possível a criação de um contador de tempo, com praticidade e facilidade de manipulação, que pode ser muito útil em aplicações hipermídia. Também é possível utilizar os algoritmos descritos e adaptá-lo para outras linguagens de programação. Ressalte-se o fato de que este contador de tempo está baseado no clock interno do computador, assim, ele define um marcador de tempo real, independente da aplicação em si e não necessita de escalas

de conversão. Essa característica torna-o uma poderosa ferramenta para simulações de movimentos e medidas de intervalos de tempos em experimentos computacionais. Deixa-se, portanto, a indicação da exploração dessa característica em trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

ASYMETRIX 1999. **Toolbook II 7.1 (build 7.10.329)**. Asymetrix Learning System, Inc.

CORTÊS, P. L. **Conhecendo e trabalhando com o toolbook**: asymetrix multimedia toolbook. São Paulo, SP: Érica, 1997.