

-
-
-
-
-
-
-
-
-

Protótipo de editor fluxogramático
com *interface* visual para geração de
código para o microcontrolador
PIC16C84 da *Microchip Technology*



Acadêmico: Drayton Roberto Fontanive
Orientador: Miguel Alexandre Wisintainer



-
-
-
-
-
-
-
-

•
•
•

Roteiro da Apresentação

- Introdução
- Fundamentação teórica
- Especificação, implementação e testes
- Conclusão

• • • • • • • •

•
•
•

Motivação

- Várias formas de uso dos microcontroladores
- Sistemas de alta performances com custo e tamanho reduzido
- Pode ser reprogramado milhares de vezes
- Aprendizado reaplicável
- Melhor visualização do problema/solução através de fluxogramas

• • • • • • • •

•
•
•

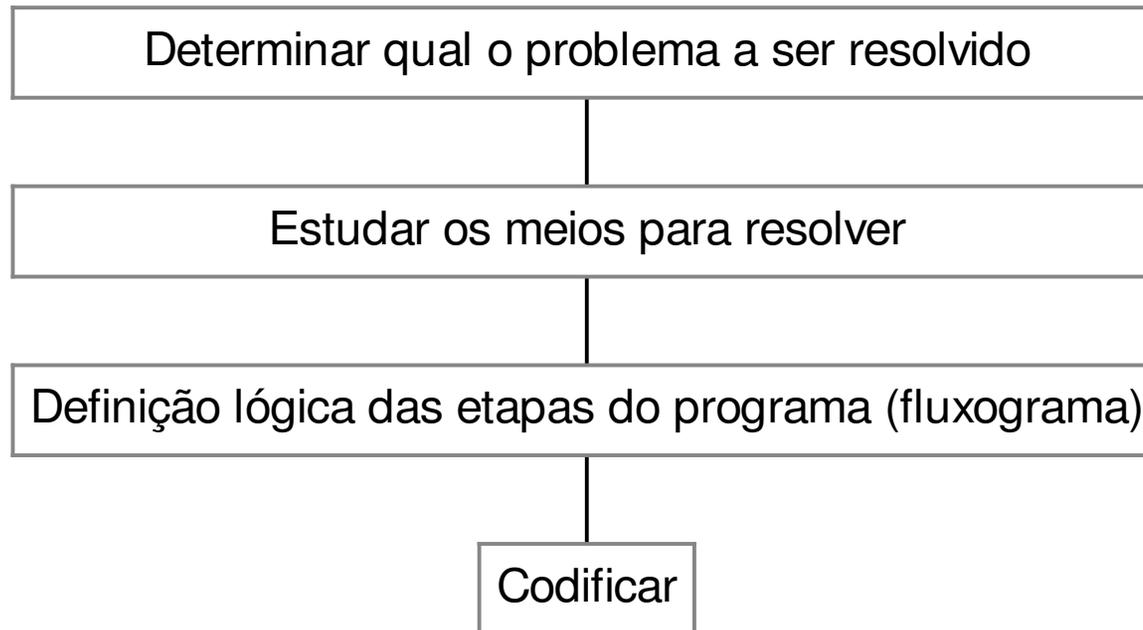
Objetivos

- Especificar e implementar um protótipo
- Breve estudo da representação das informações de forma visual
- Estudo do microcontrolador PIC16C84
- Estudo do componente *Ttree*

-
-
-

Fluxogramas

Etapas para fazer um programa



•
•
•

Codificando significados visuais

- Escolha de um símbolo gráfico para um conceito
- Escolha de uma característica gráfica
- Escolha de uma função escalar

•
•
•

Por que gráficos?

- auxiliam na performance do trabalho
- tornam documentos uma linguagem global
- aumentam a credibilidade
- são compactos
- permitem uma visão mais clara dos erros
- visão é nosso sentido predominante

• • • • • • • •

-
-
-



Microcontroladores



-
-
-
-
-
-
-
-

•
•
•

PIC (*Peripheral Interface Controller*)

- Controlador de entrada e saída
- Baseado na Arquitetura Harward
- Tecnologia RISC

• • • • • • • •

•
•
•

PIC16C84

- Possui 18 pinos = 13 são pinos de I/O
- Tecnologia RISC
- Memória de programa do tipo EEPROM
- Memória RAM

• • • • • • • •

•
•
•

TeeTree

- Objeto orientado à árvore
- Pode ser utilizado para representar qualquer organização de dados de forma hierárquica
- É pequeno e consome pouca memória
- Classes chaves: *tree*, *shapes* e conexões

• • • • • • • •

•
•
•

Propriedades e Métodos

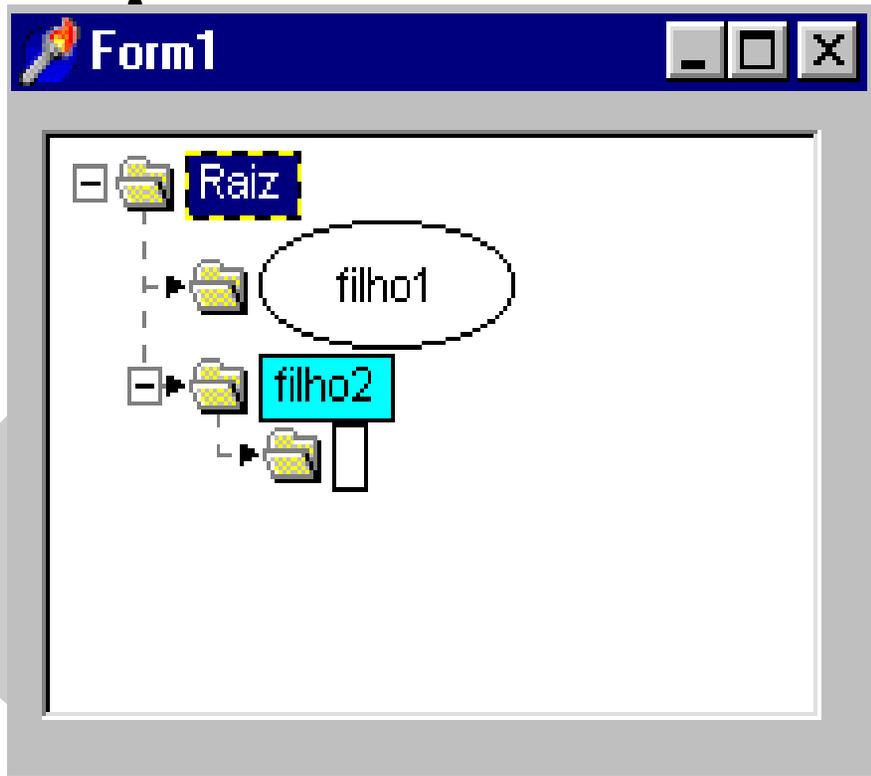
- Propriedade *Connections*
- Propriedade *Selected*
- Método *AddRoot*
- Propriedades *X0*, *X1*, *Y0*, *Y1*, *XCenter*, *YCenter*
- Propriedade *Brush.Color*

•
•
•

Propriedades e Métodos (cont.)

- Propriedade *Style*
- Propriedade *Childs*
- Método *AddChild*
- Método *AddConnections*
- Método *MoveRelative*

Aplicação Exemplo



```
procedure TForm1.FormActivate(Sender:  
TObject);
```

```
var
```

```
    filho2:ttreenodeshape;
```

```
begin
```

```
    Tree1.AddRoot('Raiz');
```

```
    Tree1[0].expanded:=true;
```

```
    Tree1[0].AddChild('filho1');
```

```
    filho2:=Tree1[0].AddChild('filho2');
```

```
    Tree1[0].Childs[0].Style:=tsscicle;
```

```
    filho2.AddChild("");
```

```
    filho2.Brush.Color:=claqua;
```

```
end;
```

•
•
•

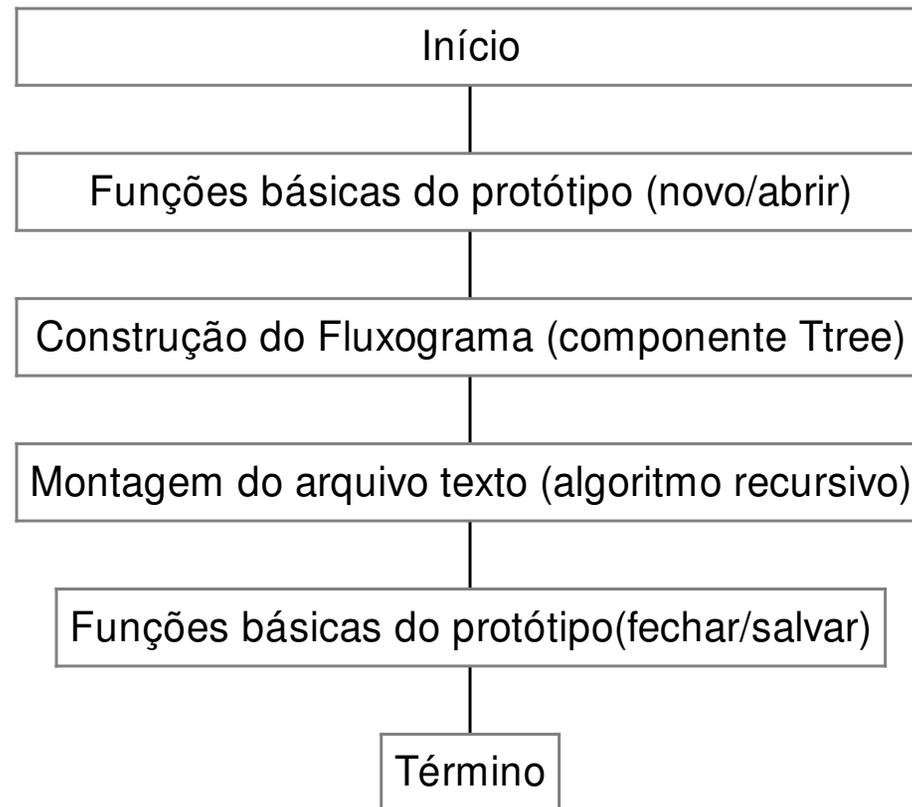
Especificação

- Ambiente de programação Delphi 3
- Componente Ttree
- Conceitos de representação visual (fluxogramas)
- Técnica de pesquisa em árvores

• • • • • • • •

•
•
•

Especificação e Operação do Protótipo



-
-
-

Inserção de novo processo



-
-
-
-
-
-
-
-
-

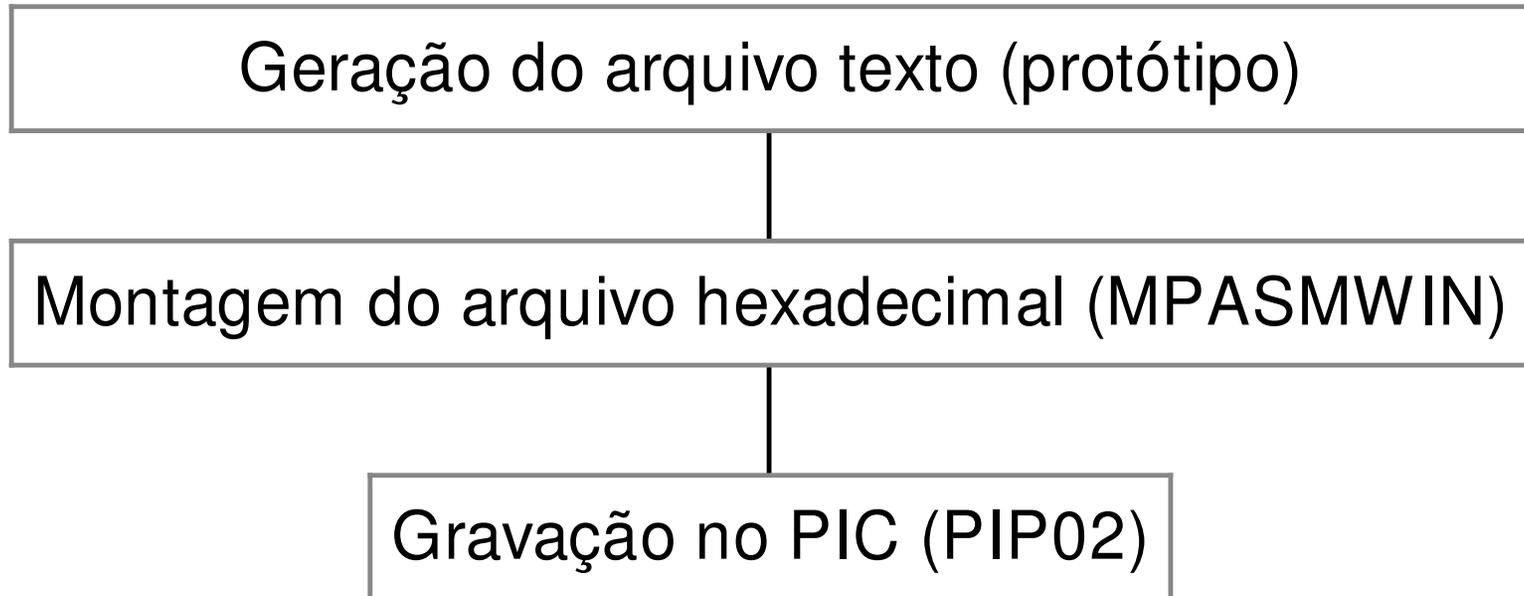
•
•
•

Testes (considerações)

- Gera apenas o que está descrito no fluxograma
- Independente da seqüência, adiciona *labels* as condições verdadeiras e falsas
- Utiliza valores decimais enquanto o *default* do assembler é hexadecimal
- O programador deve conhecer a utilidade dos registradores TRIS e PORT

-
-
-

Seqüência de operação p/ testes



-
-
-

Conclusão

- Testes realizados por *beta testers*
- Fácil e agradável de trabalhar
- Baixo consumo de memória e tamanho reduzido
- Ferramenta de ensino
- Sem erros na montagem do programa
- Possibilidade de migrar para outros microcontroladores

-
-
-

Extensões

- Implementar funções de chamadas de subrotinas e interrupções

-
-
-



FIM



-
-
-
-
-
-
-
-