



# **PROTÓTIPO DE LABORATÓRIO DE EXPERIMENTAÇÃO REMOTA MULTIPLATAFORMA CLIENTE**

**ACADÊMICO: NADER ZANOTTO  
ORIENTADOR: MIGUEL ALEXANDRE WISINTAINER**

# ROTEIRO

- Introdução
- Objetivos
- Fundamentação teórica
- Desenvolvimento do trabalho
- Conclusões
- Extensões

# INTRODUÇÃO

- Ensino a distância
  - Quebra das barreiras geográficas
  - Acesso a informação e tecnologia
- Laboratório de experimentação remota
  - Permite obter informações reais de recursos sem a necessidade de possuí-los

## OBJETIVOS

- possibilitar o desenvolvimento de software na linguagem Assembly em qualquer plataforma
- executar experimentos reais com o microcontrolador 89C51 sem a necessidade de dispor do hardware

## LABORATÓRIO DE EXPERIMENTAÇÃO REMOTA

- conjunto de instrumentos ligados a computadores os quais estão conectados a Internet (WISINTAINER, 1999, p. 32)
- vantagens
  - Experimentos reais
  - Custos hardware
  - Redução de riscos

# MICROCONTROLADOR 80C51

- Características
  - CPU 8 bits, 64kB end. dados, 64kB end. programa, 4kB mem. prog. e 128kbits de mem. dados, 32 linhas end. E/S, 5 int., 1 oscilador
- Estrutura para execução de um programa
  - barramentos dados, endereço, controle
  - memória interna 0000H->1FFFH.
  - memória externa
  - oscilador
- Execução de um programa
  - reset
  - memória interna ou externa (EA – baixo externa, alto interna)
  - Registrador PC

# INTERAÇÃO COM O BARRAMENTO DO COMPUTADOR

- Conectar periféricos
- Tipos conhecidos
  - ISA, PCI, AGP...
- Barramento ISA
  - 8 bits, 16 bits
- Principais pinos
  - A2 - A9: dados
  - A12 – A31: endereços
  - B1: GROUND
  - B3: +5V
  - B9: +12V

## APLICAÇÃO CLIENTE / SERVIDOR

- Abordagem da computação que separa processos em plataformas independentes e que permitem o compartilhamento de recursos obtendo o máximo de benefícios (BOCHENSKI, 1995, p. 8).
- Características
  - operacionalidade em rede
  - ações iniciam no cliente
  - servidor e cliente podem operar em diferentes plataformas



## APLICAÇÃO EM AMBIENTE WEB

- Evolução de *sites*
- Permite ao usuário executar uma regra do negócio através de um navegador (CONALLEN, 2003, p.11).
- Gerenciamento do estado do cliente
  - *cookies*: fragmento de dado
- Tecnologias de ativação
  - *scripts*
  - *JSP, ASP, PHP*

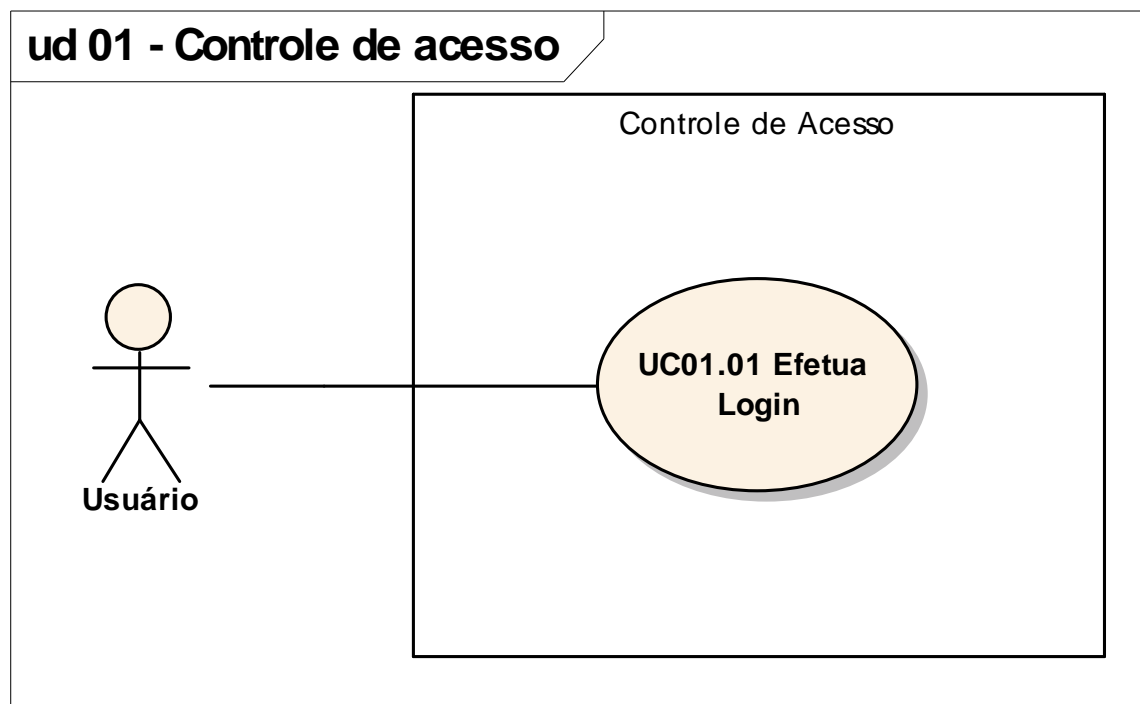
## PRINCIPAIS REQUISITOS DA IMPLEMENTAÇÃO

- Fornecer uma área para digitação do código a ser traduzido
- Permitir acompanhar o experimento verificando o conteúdo de registradores e posições de memória
- Chamar um montador para verificar o código recebido

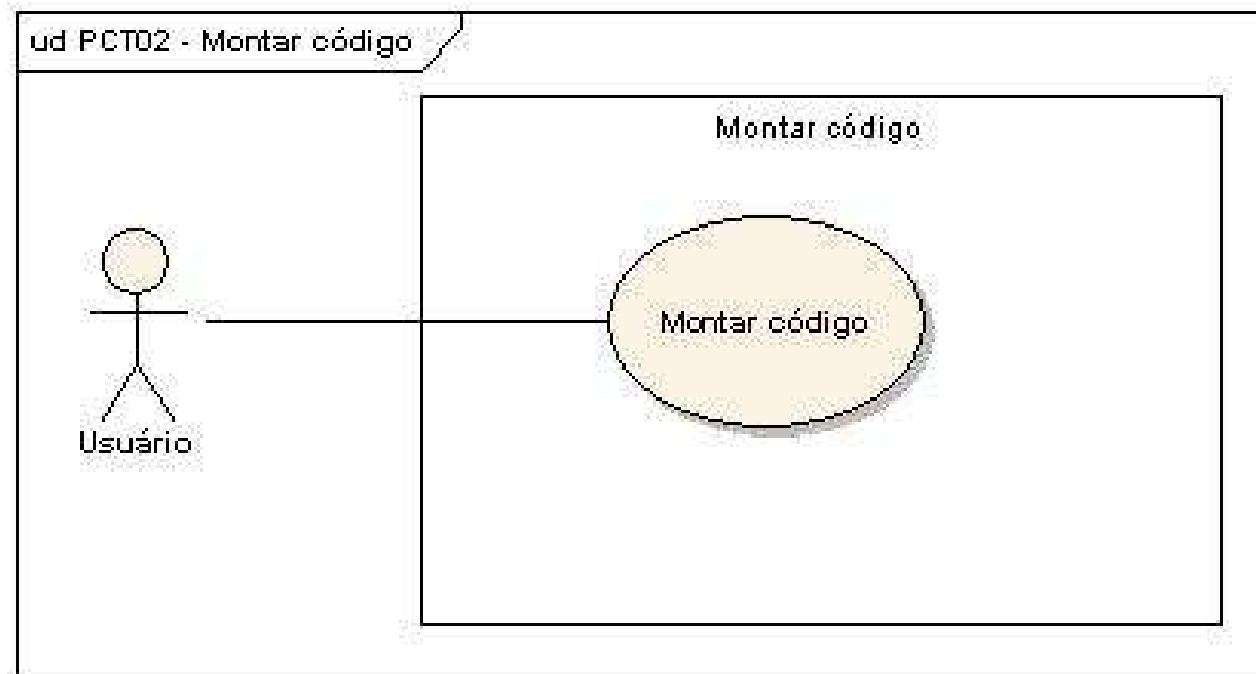
# ESPECIFICAÇÃO

- UML para especificação
- Enterprise Architect
- Diagramas de caso de uso, de atividades e classe

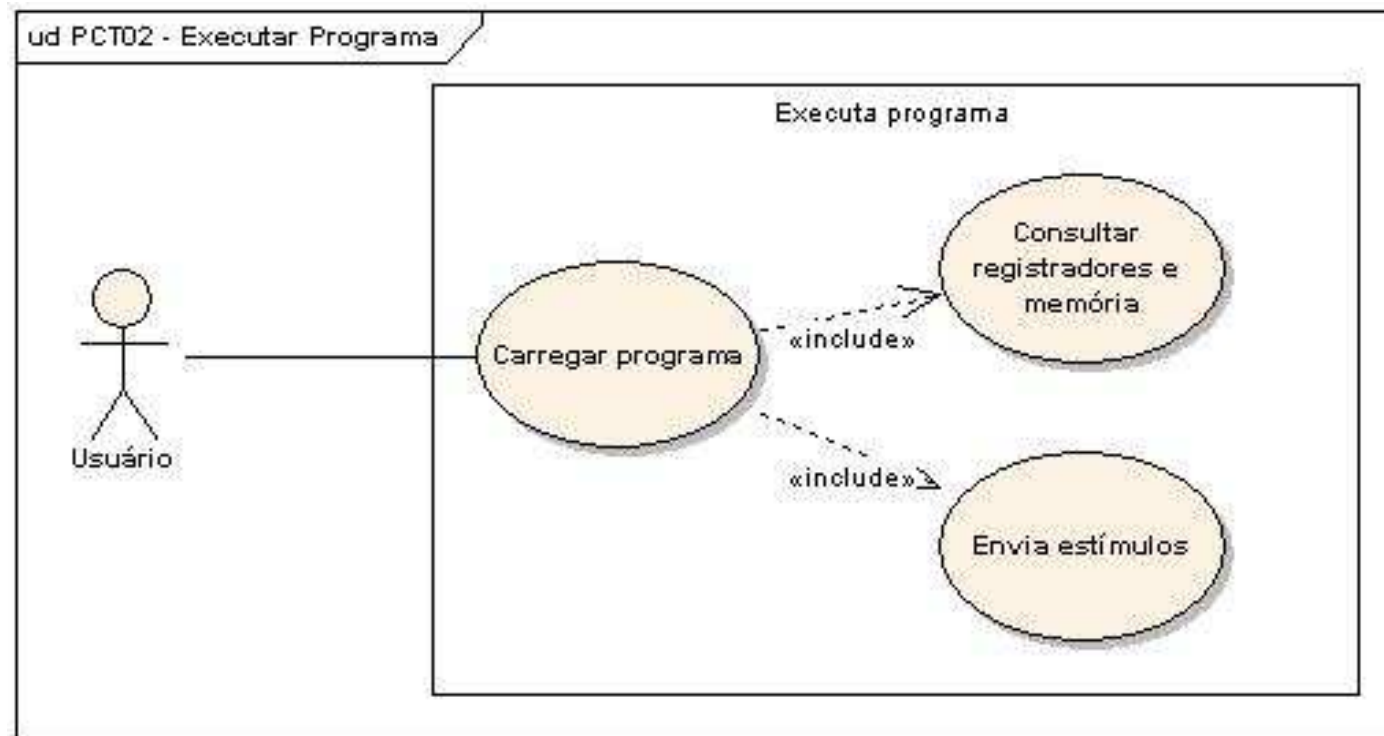
# CASO DE USO – Controle de Acesso



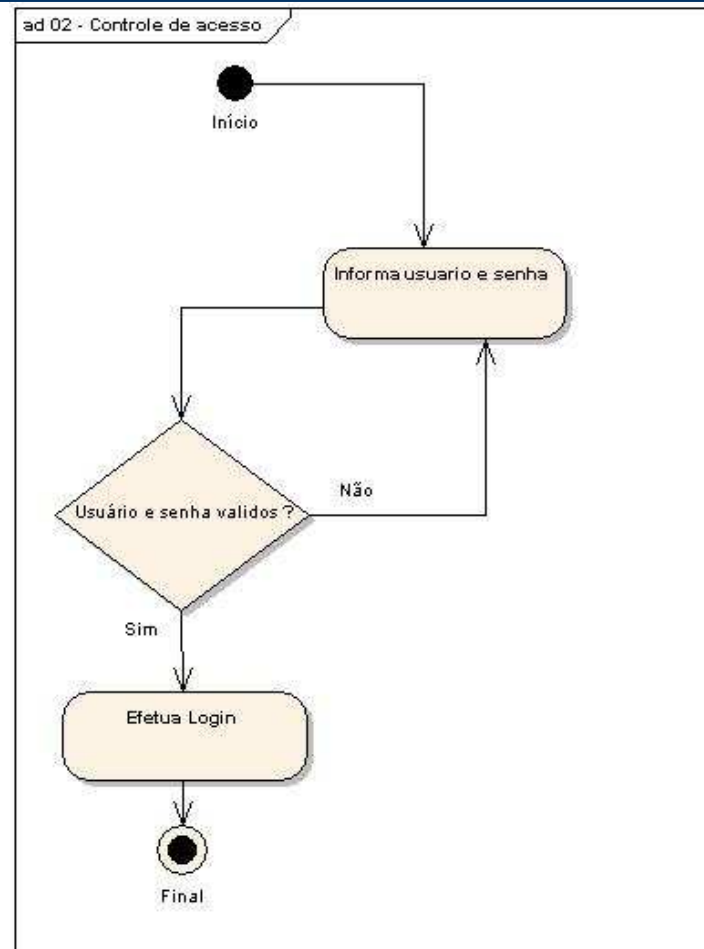
# CASO DE USO – Montar Código



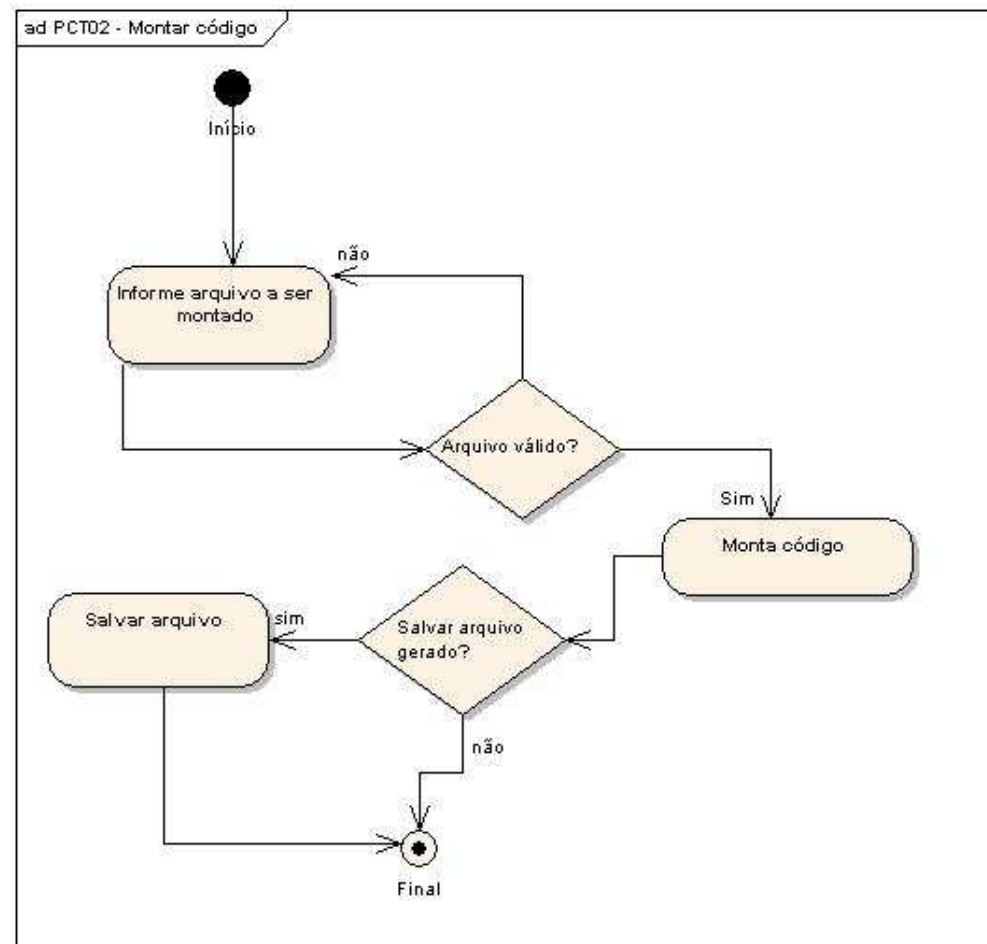
# CASO DE USO – Executar Programa



# DIAGRAMA DE ATIVIDADES – Controle de Acesso

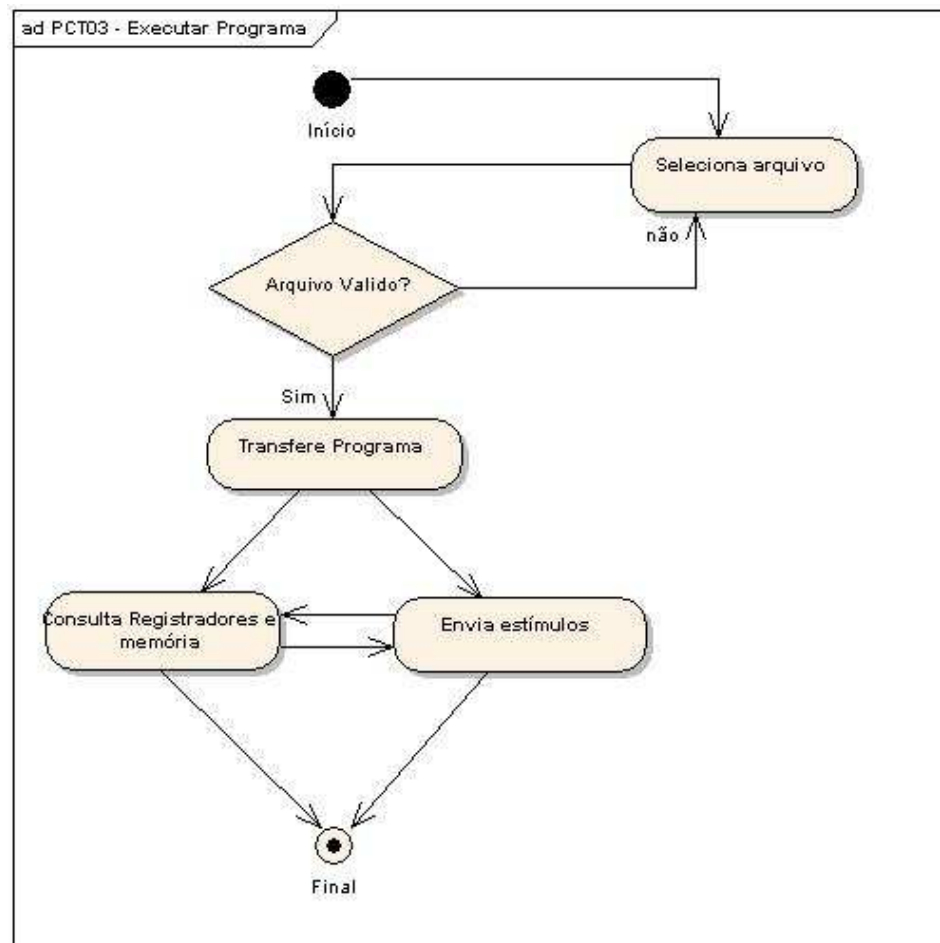


# DIAGRAMA DE ATIVIDADES – Montar Código

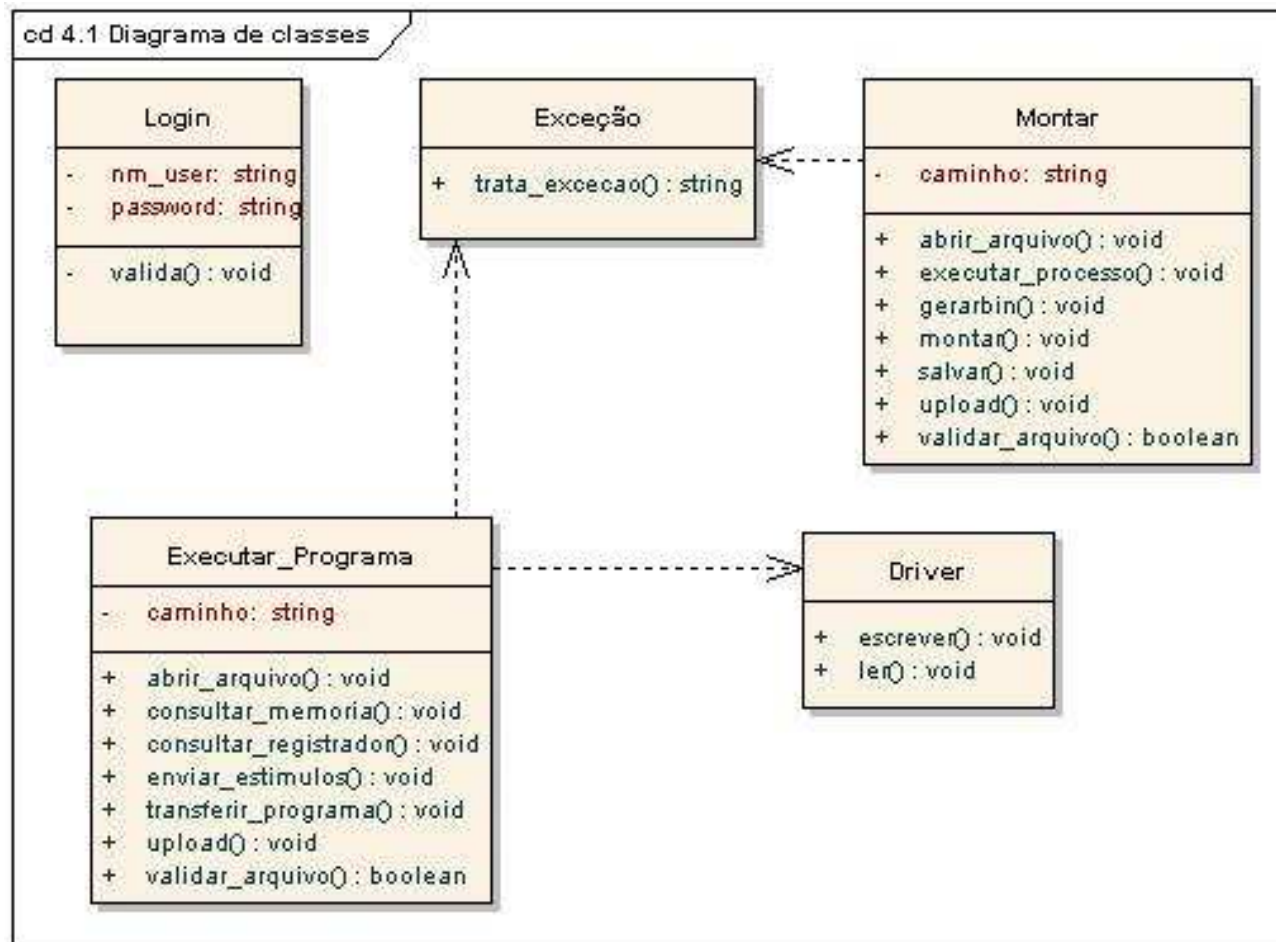




# DIAGRAMA DE ATIVIDADES – Executar Programa



# DIAGRAMA DE CLASSES



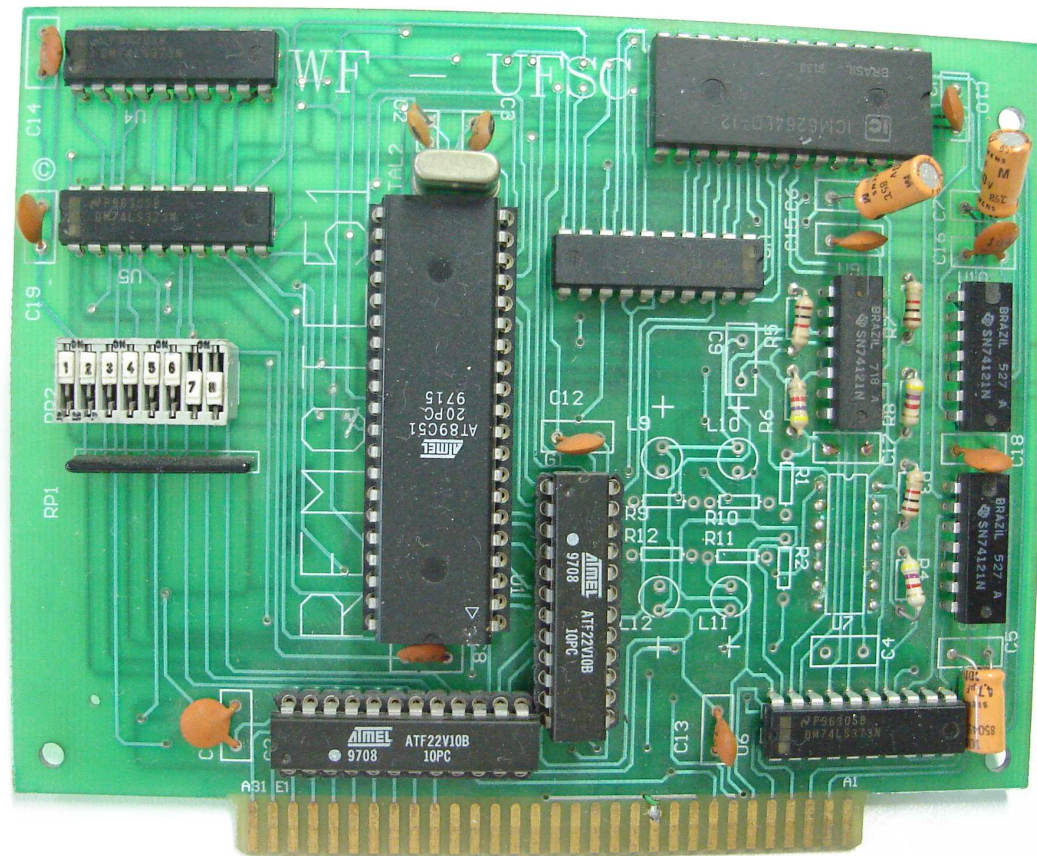
## FERRAMENTAS UTILIZADAS

- Microsoft Visual Studio 2003
- ASM51
- RAPIDDRIVER
- Placa com 89C51

## A PLACA DO MICROCONTROLADOR

- Desenvolvida por Wisintainer
- Componentes da placa : microcontrolador 89C51, 4 *buffers*, 1 memória 6224, 3 monoestáveis 74121, 2 GAL , 1 chave para endereçamento

# A PLACA DO MICROCONTROLADOR



## **DRIVE PARA COMUNICAÇÃO COM A PLACA**

- **RAPIDDRIVER (Entech)**
- **Características: Trial, criação de projetos, suporte a diferentes linguagens, etc.**

# CÓDIGO DE TRANSFERÊNCIA DO PROGRAMA DO USUÁRIO

```
private void TransferCode()
{
    //carrega arquivo .BIN e efetua a leitura a partir do endereço 0x8000
    FileStream f = new FileStream(localpath + session["filename"], FileMode.Open);
    f.Seek(0x8000, SeekOrigin.Begin);
    LblMsg.Text = "";
    //abrindo Driver (dll)
    driveropen();
    RapidIsa_Cs.RapidIsa.OpenRapidIsa(0);
    //detectando a placa do microcontrolador
    RapidIsa_Cs.RapidIsa.SetPortByte(hIsa, Address1, 0);
    Thread.Sleep(200);
    LblMsg.Text = RapidIsa_Cs.RapidIsa.GetPortByte(hIsa, Address0).ToString();
    Thread.Sleep(time);
    RapidIsa_Cs.RapidIsa.SetPortByte(hIsa, Address1, 0);
    Thread.Sleep(time);
    //informando o tamanho do programa
    int len = Convert.ToInt32(f.Length-0x8000);
    //dividindo o tamanho em 2 bytes
    byte ms = Convert.ToByte(len/256);
    byte ls = Convert.ToByte(len%256);
    //enviando o byte mais significativo
    RapidIsa_Cs.RapidIsa.SetPortByte(hIsa, Address0, ms);
    Thread.Sleep(time);
    //enviando o byte menos significativo
    RapidIsa_Cs.RapidIsa.SetPortByte(hIsa, Address0, ls);
    Thread.Sleep(time);
    //transferindo o programa
    while (f.Position < f.Length)
    {
        byte comando = Convert.ToByte(f.ReadByte());
        RapidIsa_Cs.RapidIsa.SetPortByte(hIsa, Address0, comando);
        Thread.Sleep(time);
    }
    //mandando o "lixo" para informar final de programa
    RapidIsa_Cs.RapidIsa.SetPortByte(hIsa, Address0, 0);
    Thread.Sleep(time);
    LblMsg.Text = "Programa Carregado com sucesso!";
    f.Close();
}
```

# OPERACIONALIDADE

## Bem Vindos

Protótipo de Laboratório de Experimentação Remota  
com  
Microcontrolador 89C51

Usuário

Senha

Login

Protótipo desenvolvido pelo Acadêmico Nader Zanotto.



# OPERACIONALIDADE

Protótipo de Laboratório de Experimentação Remota – Microcontrolador 89C51

[Montador](#)

**Conteúdo dos Registradores**

**Status:**

**Estímulos**

P1.0  
 P1.1  
 P1.2  
 P1.3  
 P1.4  
 P1.5  
 INT0  
 T0

**Memória**

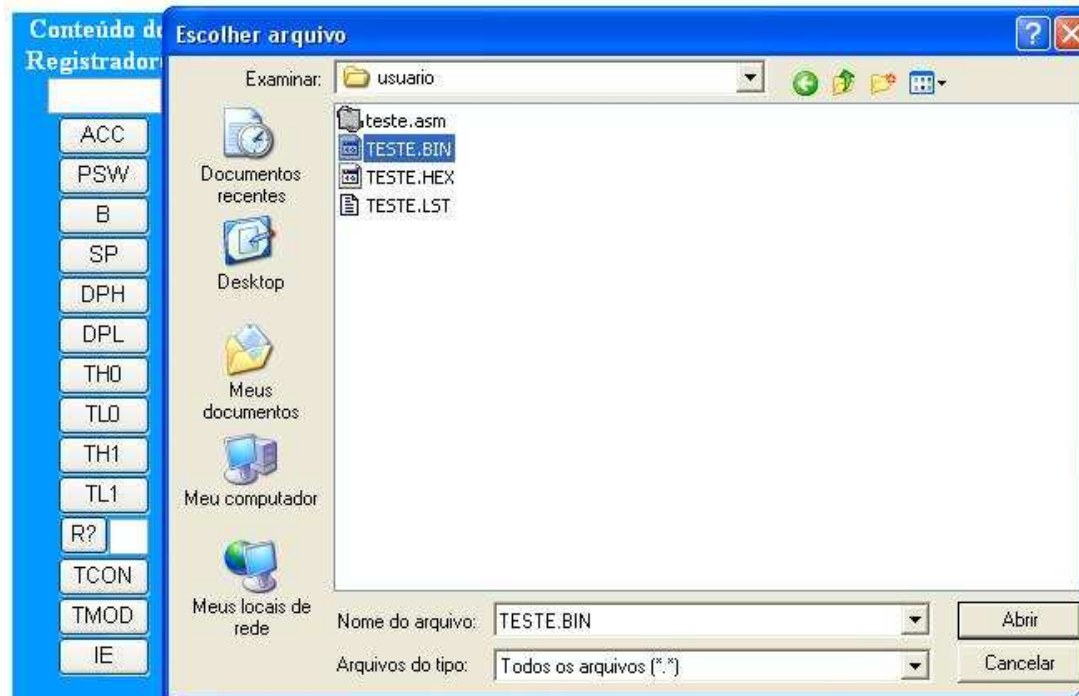
Memória

[Ajuda?](#)  
[Envie seu Comentário](#)

# OPERACIONALIDADE

Protótipo de Laboratório de Experimentação Remota – Microcontrolador 89C51

Procurar... Carregar Montador



ajuda?  
crie seu  
diagnóstico

# OPERACIONALIDADE

Protótipo de Laboratório de Experimentação Remota – Microcontrolador 89C51

Procurar... Carregar [Montador](#)

Programa Carregado com sucesso!

Status:

**Conteúdo dos Registradores**

55

ACC

PSW

B

SP

DPH

DPL

TH0

TL0

TH1

TL1

R?

TCON

TMOD

IE

**Estímulos**

P1.0

P1.1

P1.2

P1.3

P1.4

P1.5

INT0

T0

Enviar

**Memória**

Memória

[Ajuda?](#)

[Envie seu Comentário](#)

**Visualiza valor do registrador selecionado**

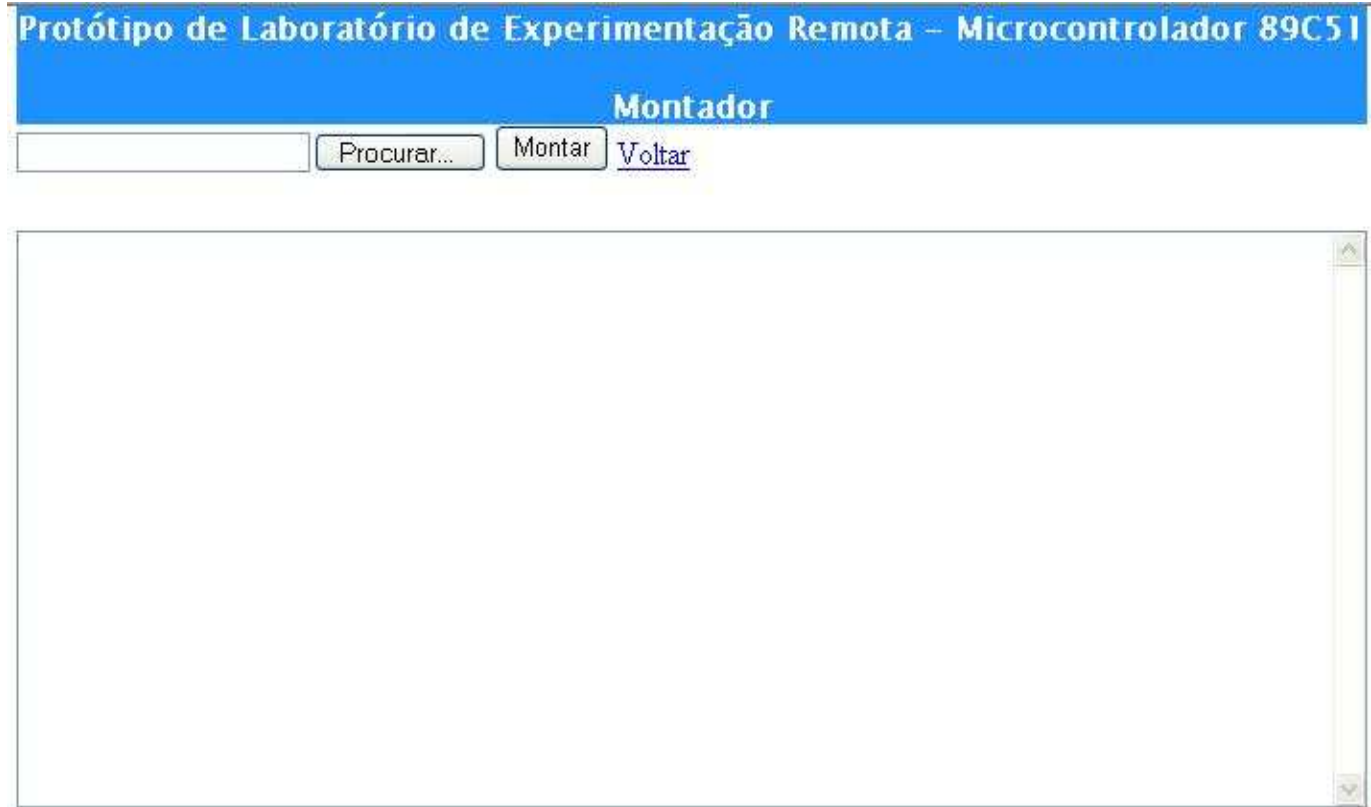
**Seleciona registrador**

**Seleciona Estímulos**

**Seleciona posição de memória**

**Visualiza valor da posição de memória**

# OPERACIONALIDADE



# OPERACIONALIDADE

Protótipo de Laboratório de Experimentação Remota – Microcontrolador 89C51

Montador

C:\asm51\teste3.asm

Procurar...

Montar

Voltar

Download de Arquivo

Deseja salvar este arquivo?



Nome: teste3.Bin

Tipo: Tipo de Arquivo Desconhecido, 32,0 KB

De: 192.168.254.22

Salvar

Cancelar



Embora arquivos provenientes da Internet possam ser úteis, alguns arquivos podem danificar seu computador. Se você não confiar em sua origem, não salve este arquivo. [Qual é o risco?](#)

# OPERACIONALIDADE

Protótipo de Laboratório de Experimentação Remota – Microcontrolador 89C51

Montador

Procurar...

Montar

[Voltar](#)

```
      2
8000      3      ORG 8000H
      4
8000 7437      5          MOV A,#55
      6
8002 75F040    7          MOV B,#64
      8
8005      9      BEGIN:
      10
8005 028005   11          LJMP BEGIN
      12      erro ←
      13      END
****-----^
****ERROR #23: Illegal or missing directive
      14
      15

VERSION 1.2h ASSEMBLY COMPLETE. 2 ERRORS FOUND
```

Erro no código

## RESULTADOS E DISCUSSÕES

- No geral os objetivos foram atingidos
- Dificuldades com os *drivers* de comunicação com o barramento ISA

## CONCLUSÕES

- O Protótipo cumpriu com os objetivos
- Alcance do protótipo
- Limitações :
  - Sem um editor de texto



## EXTENSÕES

- Atualizar o barramento de comunicação da placa
  - ISA -> PCI
- Aumentar e ampliar o número de periféricos controlados