

CONSTRUÇÃO DE UM PROTÓTIPO PARA AUTOMATIZAÇÃO DA LEITURA DE NÍVEIS DE FLUMENS

Elaborado por Gari Julio Einsfeldt

Banca: Prof. Miguel Wisintainer (orientador)
Prof. Antonio Carlos Tavares
Prof. Francisco Adell Péricas

ROTEIRO

- Introdução
- Fundamentação
- Desenvolvimento do Hardware
- Desenvolvimento do Software
- Conclusões

INTRODUÇÃO

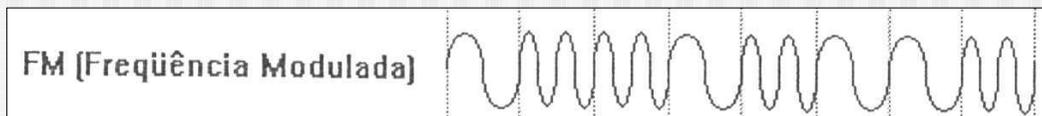
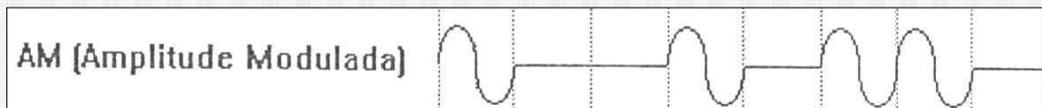
- Contextualização do Trabalho
 - Problemas da colonização humana;
 - Resultado;
 - Objetivo.
- Objetivos do trabalho

Requisitos

- Estabelecer meio de comunicação;
- Possibilitar identificação de inúmeras réguas.
- Criar regras para comunicação;
- Evitar erros;
- Leitura em frequência variada.

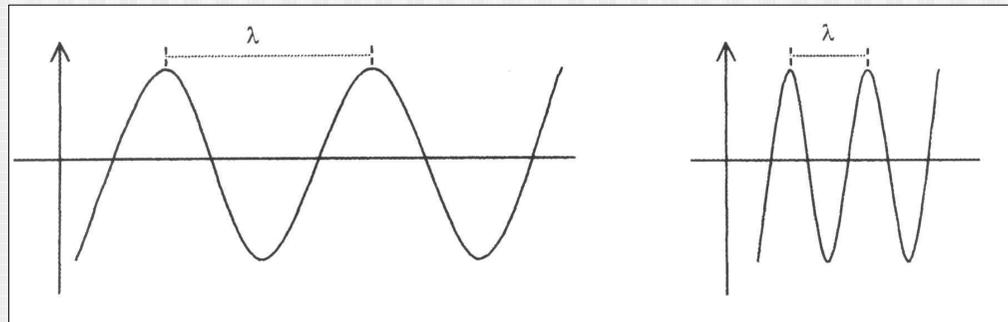
FUNDAMENTAÇÃO TEÓRICA

- Comunicação por interface
- Transmissão
- Modulação



FUNDAMENTAÇÃO TEÓRICA

- Comunicação por interface
- Transmissão
- Modulação
- Radiofrequência



FUNDAMENTAÇÃO TEÓRICA

- Comunicação por interface
- Transmissão
- Modulação
- Radiofrequência
- Protocolo

Abertura	Endereço	Dado	Consistência	Fechamento
-----------------	-----------------	-------------	---------------------	-------------------

FUNDAMENTAÇÃO TEÓRICA

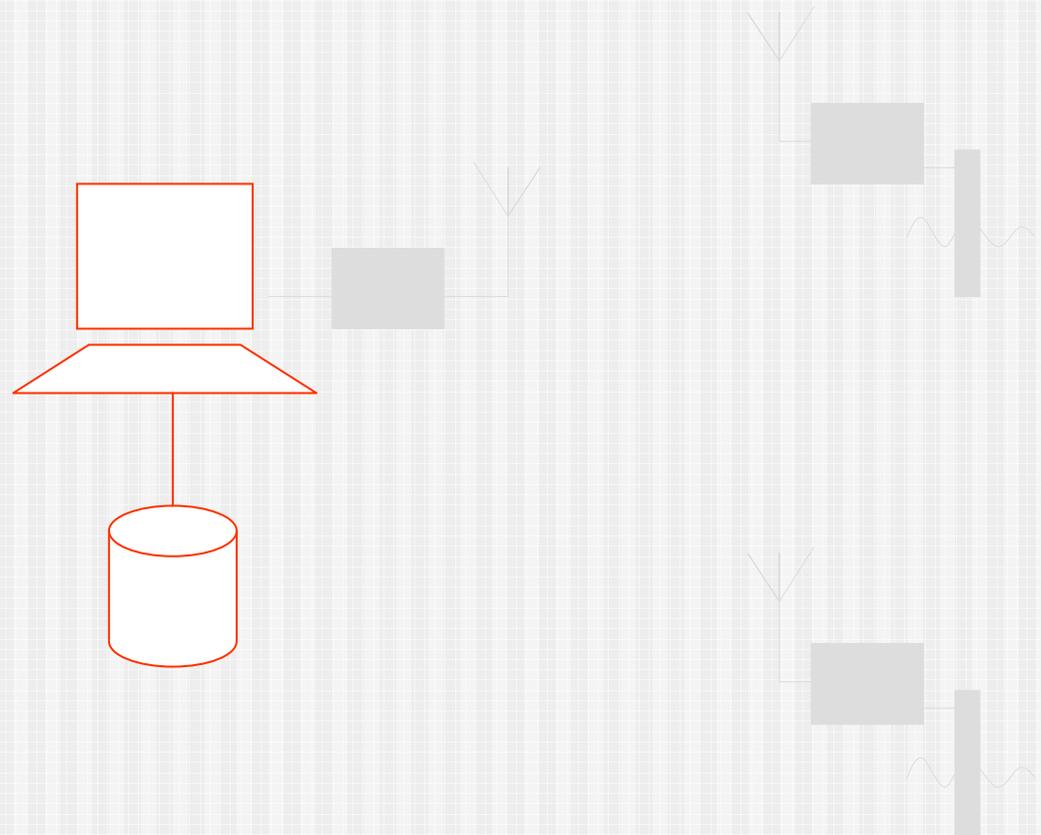
- Comunicação por interface
- Transmissão
- Modulação
- Radiofrequência
- Protocolo
- Microcontroladores
- Modelagem estruturada

Requisitos

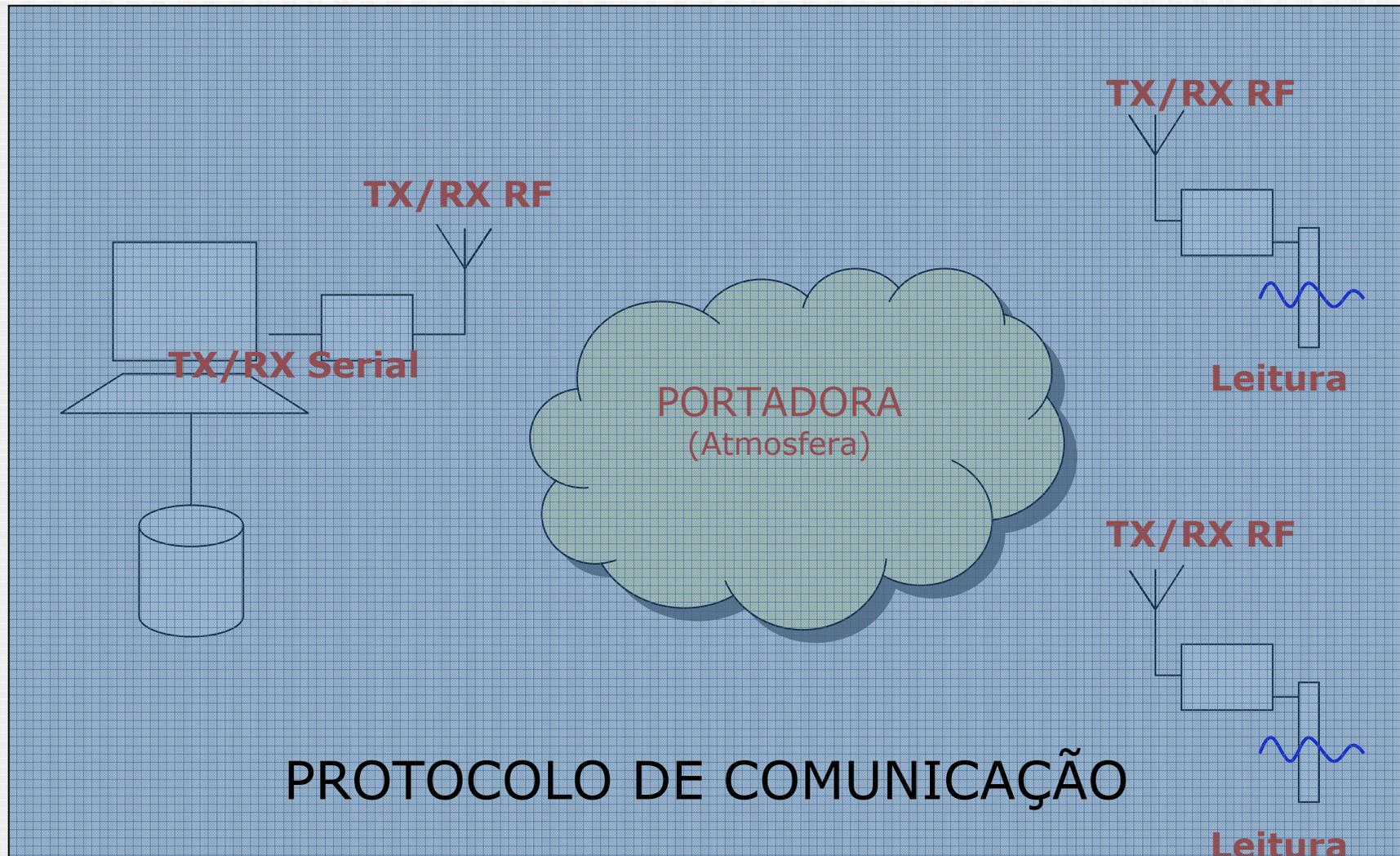
- Criar módulos de comunicação radiofreqüência:
 - Controle
 - Leitura
- Gerir a comunicação
- Armazenar coletas em banco de dados;
- Demonstrar as informações coletadas

Cenário (Software)

- Módulo de Gestão
- Módulo de Coleta

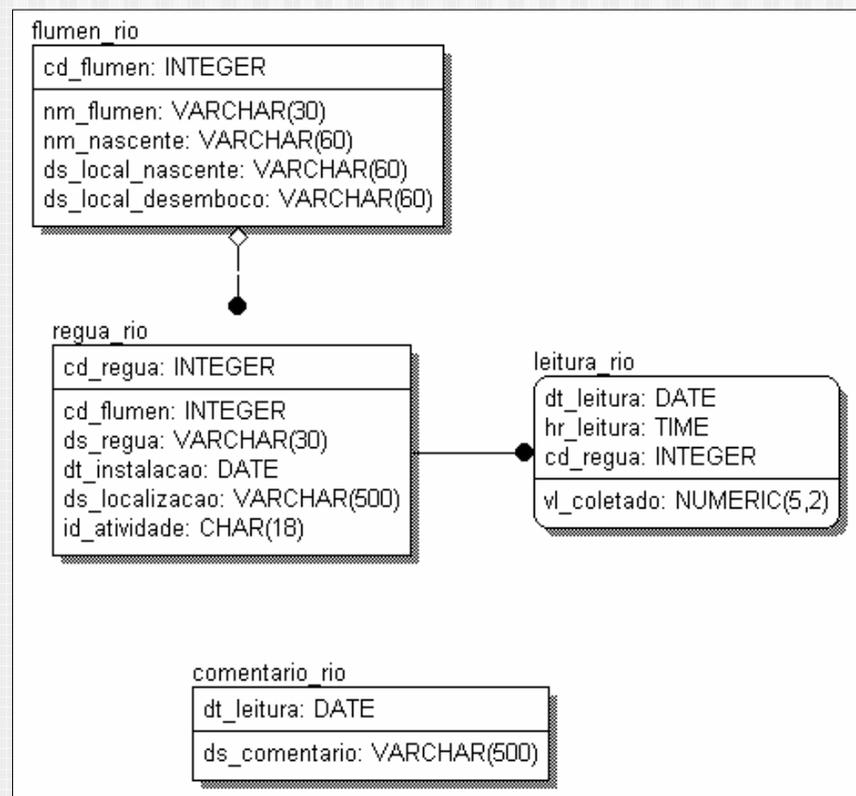


Cenário (Hardware)



ESPECIFICAÇÃO

■ Modelo entidade relacionamento



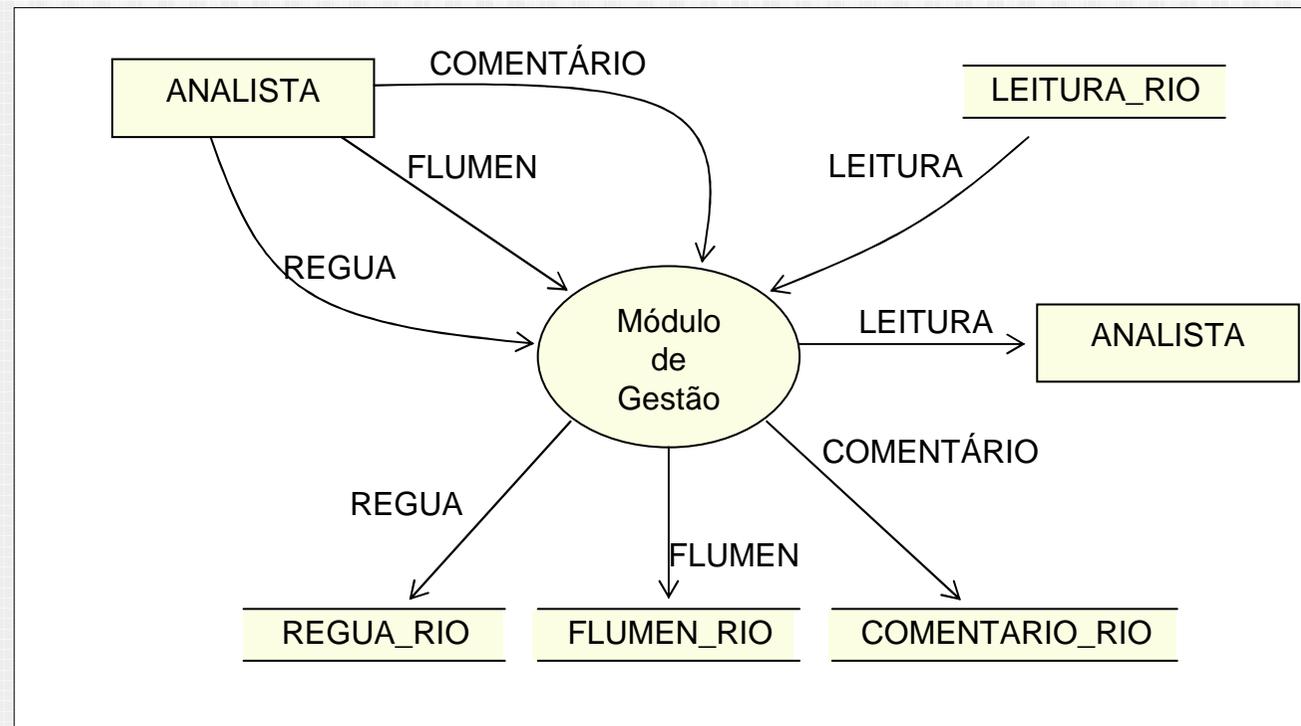
ESPECIFICAÇÃO

- Modelo entidade relacionamento
- Protocolo

ABERTURA	DADO	FECHAMENTO
1 Byte (*)	1 Byte	1 Byte (#)

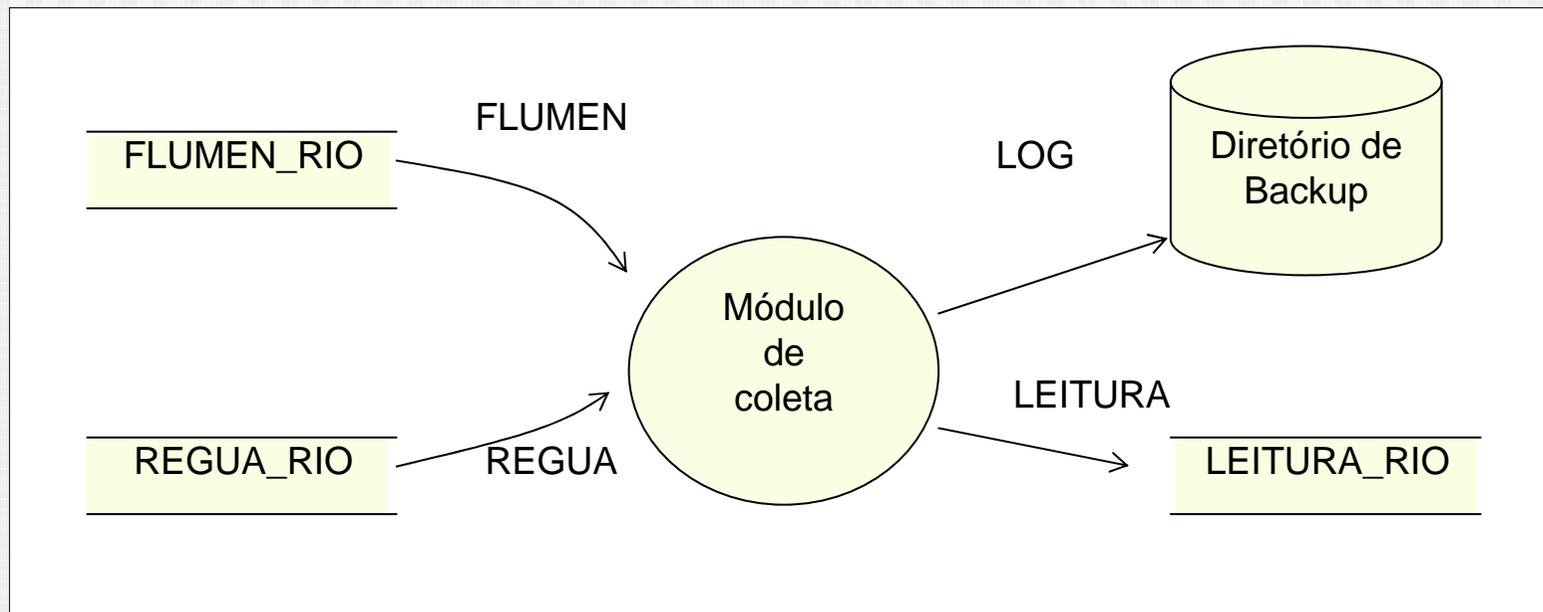
ESPECIFICAÇÃO

- Sistema de Gestão
 - DFD de Contexto



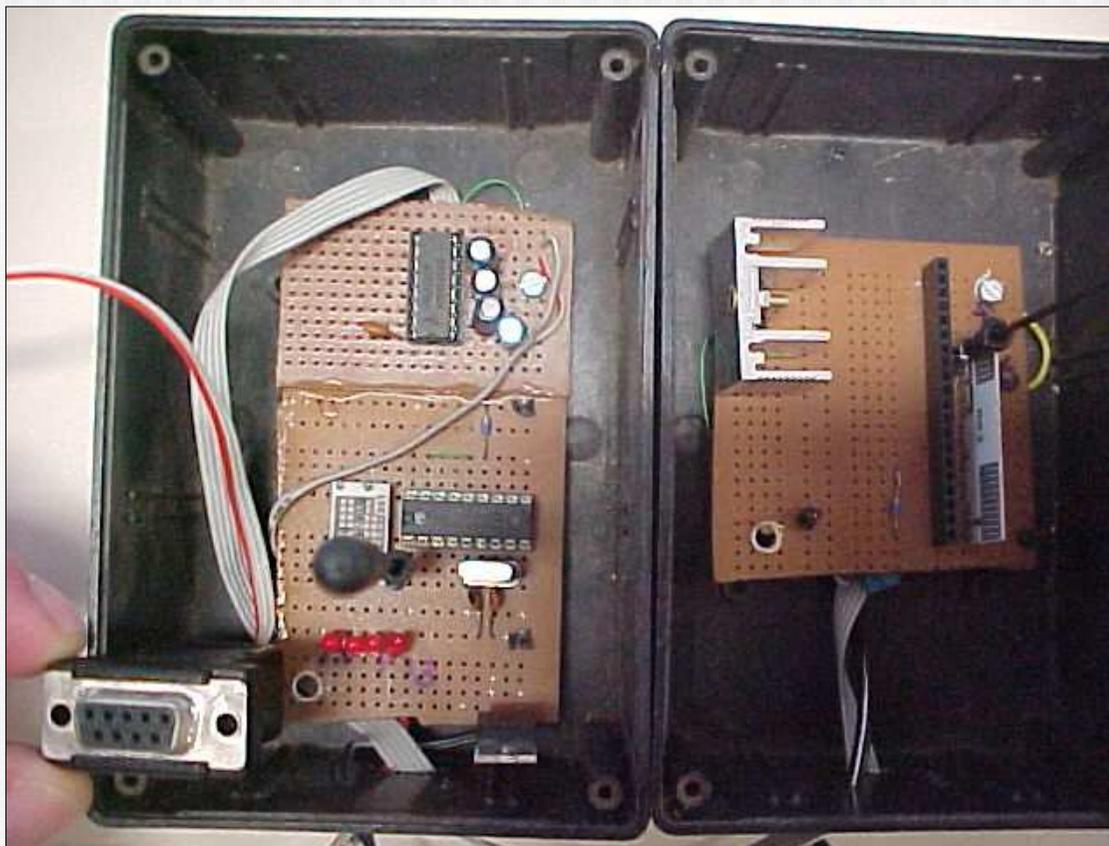
ESPECIFICAÇÃO

- Sistema de Coleta
 - DFD de Contexto



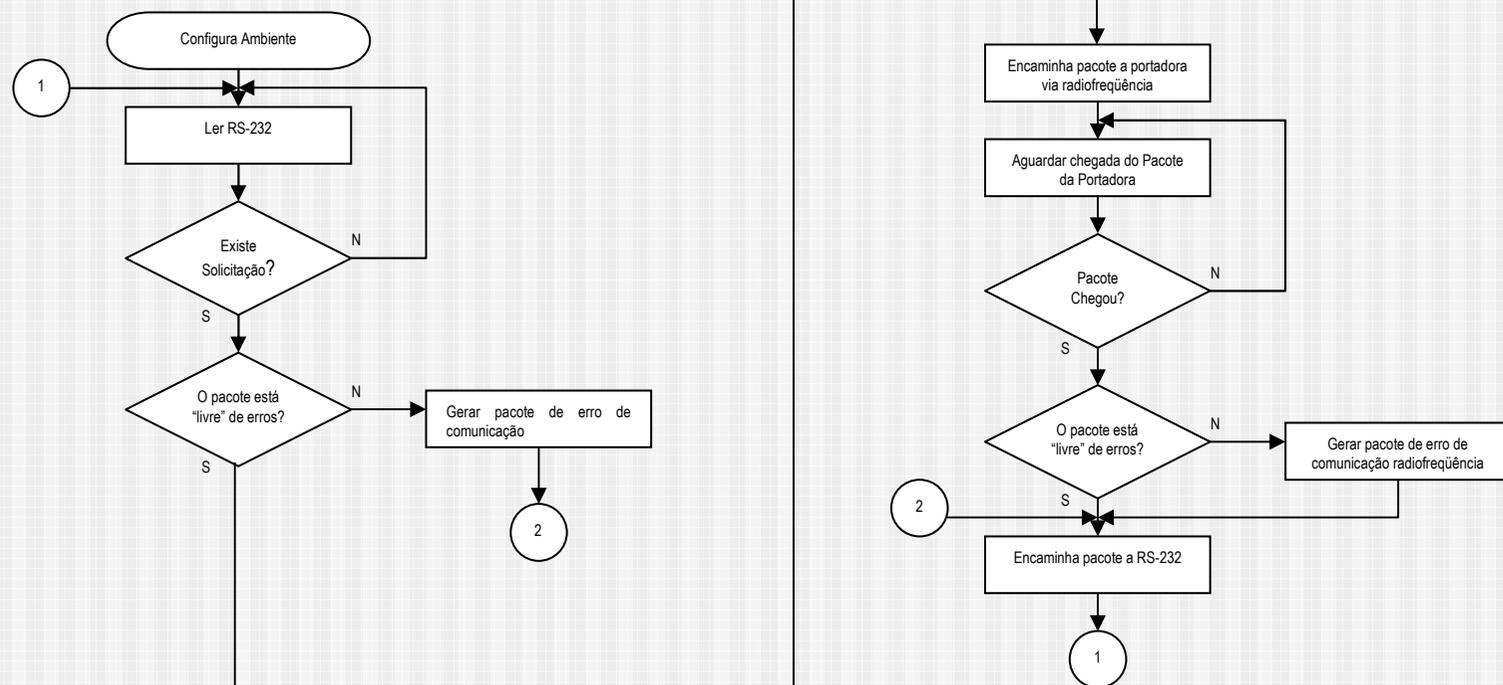
ESPECIFICAÇÃO

- Módulo de Controle



ESPECIFICAÇÃO

■ Fluxograma



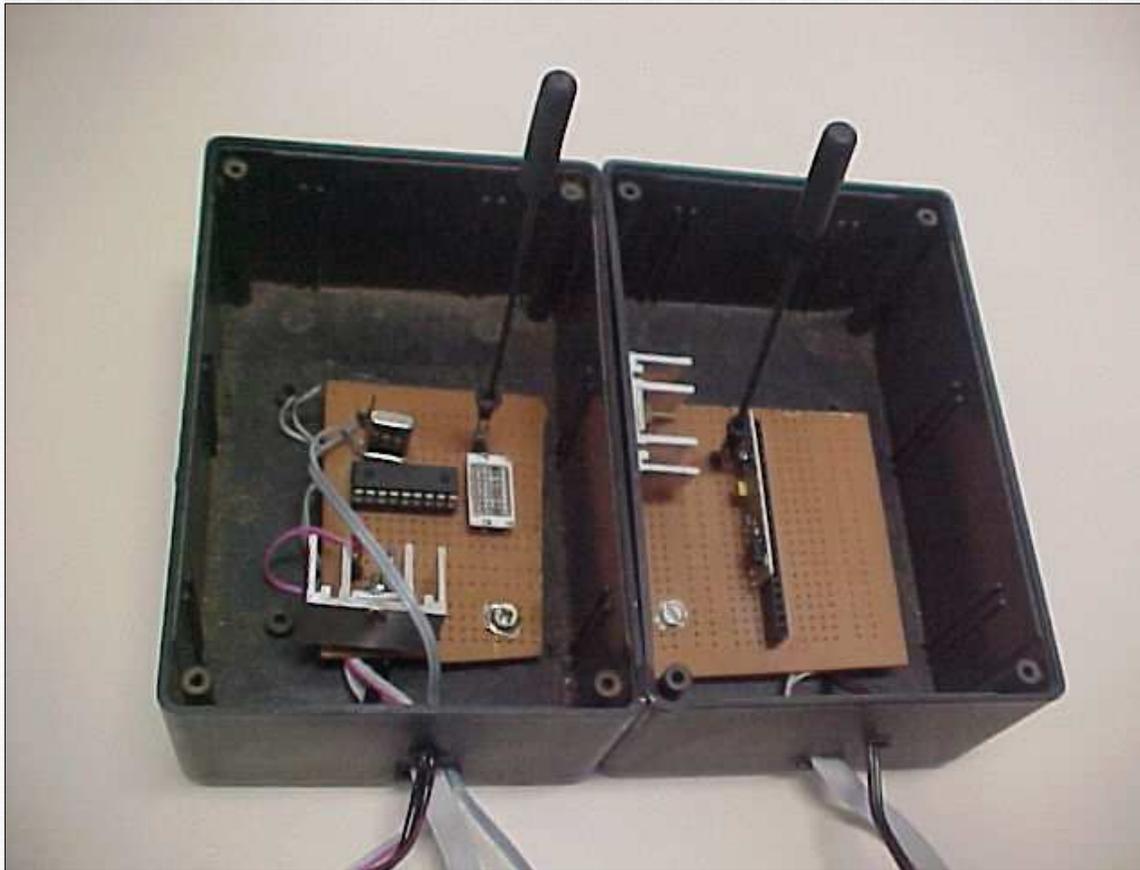
IMPLEMENTAÇÃO

```
NIVEL          VAR WORD  'Variavel que recebe o nível do Rio
BAUD           CON 16780 'Taxa de Transferencia via RF Inverted N2400 baud
BAUD_NO_INVERTED CON 396  'Taxa de Transferencia via Serial NON INVERTED
JUNK           CON 126   'Caractere desconsiderado, usado para estabilizar o sinal
SYNCH         CON "A"   'Caractere de Sincronismo
CR_SIN        CON "*"   'Caractere de inicio de pacote (START)
CR_END        CON "#"   'Caractere de termino de pacote (END)
SLAVE         VAR BYTE  'Variavel receptora do pacote
CR_SIN_REC    VAR BYTE  'Recebe caractere de inicio de pacote recebido
CR_END_REC    VAR BYTE  'Recebe caractere de final de pacote recebido
TRISA=%00000000
START:
    'Aguarda requisição vinda por porta serial RS-232 do PC
    SERIN2 PORTB.2,BAUD_NO_INVERTED,1000,CAI_FORA,[CR_SIN_REC,SLAVE,CR_END_REC]
    'Verifica se houve erro de comunicação
    IF CR_SIN_REC<>CR_SIN AND CR_END_REC<>CR_END THEN
        GOTO START
    ENDIF

    'Transmite a requisição do PC para a portadora RF
    SEROUT2 PORTB.4,BAUD,[junk,SYNCH,CR_SIN,SLAVE,CR_END]
    'Aguarda 1 segundo para receber as informações vinda das estações
    SERIN2 PORTB.5,BAUD,1000,CAI_FORA,[WAIT(SYNCH),CR_SIN_REC,NIVEL,CR_END_REC]
    'Valida pacote
    IF CR_SIN_REC=CR_SIN AND CR_END_REC=CR_END THEN
        POKE PORTA,NIVEL 'Atualiza leds com o valor recebido
        SEROUT2 PORTB.3,BAUD_NO_INVERTED,[CR_SIN,NIVEL,CR_END] 'Retorna valor ao PC
    ENDIF
    PAUSE 250
CAI_FORA:
    GOTO START
```

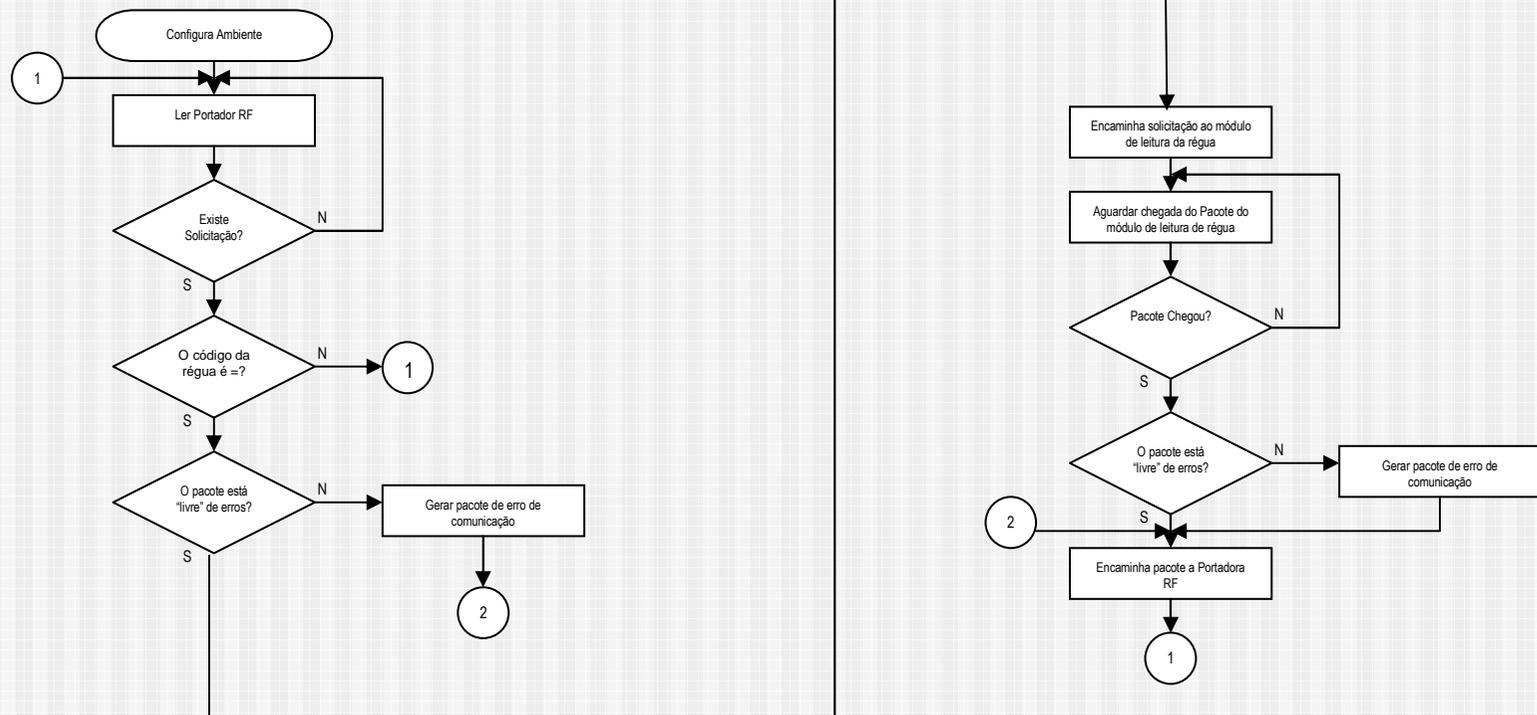
ESPECIFICAÇÃO

- Módulo de Base



ESPECIFICAÇÃO

■ Fluxograma



IMPLEMENTAÇÃO

```
NIVEL          VAR WORD  'Variavel que recebe o nível do Rio
BAUD           CON 16780 'Taxa de Transferencia via RF Inverted N2400 baud
BAUD_NO_INVERTED CON 396  'Taxa de Transferencia via Serial NON INVERTED
JUNK           CON 126   'Caractere desconsiderado, usado para estabilizar o sinal
SYNCH         CON "A"   'Caractere de Sincronismo
CR_SIN        CON "*"   'Caractere de inicio de pacote (START)
CR_END        CON "#"   'Caractere de termino de pacote (END)
SLAVE         VAR BYTE  'Variavel receptora do pacote
SLAVE_ID      CON 1     'Identificador do circuito
CR_SIN_REC    VAR BYTE  'Recebe caractere de inicio de pacote recebido
CR_END_REC    VAR BYTE  'Recebe caractere de final de pacote recebido
TRISA=%00000010
START:
  'Aguarda requisição vinda por porta serial RS-232 do PC
  SERIN2 PORTB.5,BAUD,[WAIT(SYNCH),CR_SIN_REC,SLAVE,CR_END_REC]
  'Verifica se houve erro de comunicação
  IF CR_SIN_REC<>CR_SIN AND CR_END_REC<>CR_END THEN
    GOTO START
  ENDIF

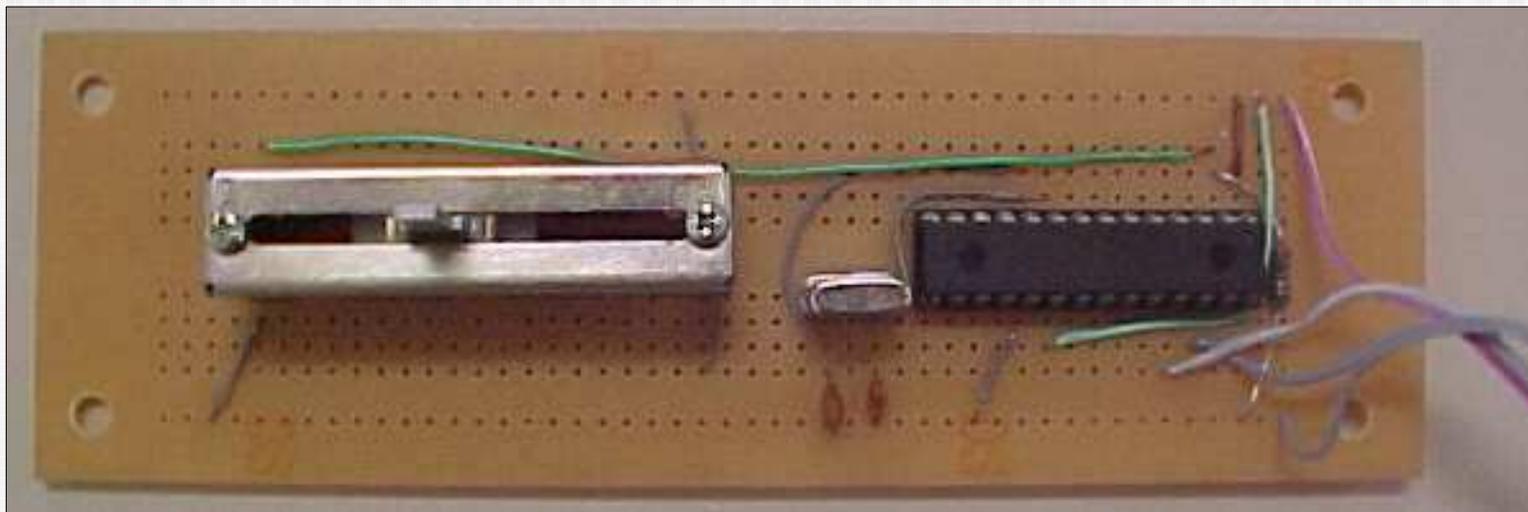
  'VERIFICA SE O PACOTE EH PARA MIM
  IF SLAVE=SLAVE_ID THEN
    'AQUI SINALIZA PIC16F876 PARA LER A/D
    POKE PORTA,255

    'AQUI ESPERA DADO PELA SERIAL DO PIC16F873
    SERIN2 PORTA.1,BAUD,700,CAI_FORA,[CR_SIN_REC,NIVEL,CR_END_REC]

    'ENCAMINHA LEITURA
    SEROUT2 PORTB.4,BAUD,[JUNK,SYNCH,CR_SIN,NIVEL,CR_END]
  ENDIF
CAI_FORA:
  POKE PORTA,0
  GOTO START
```

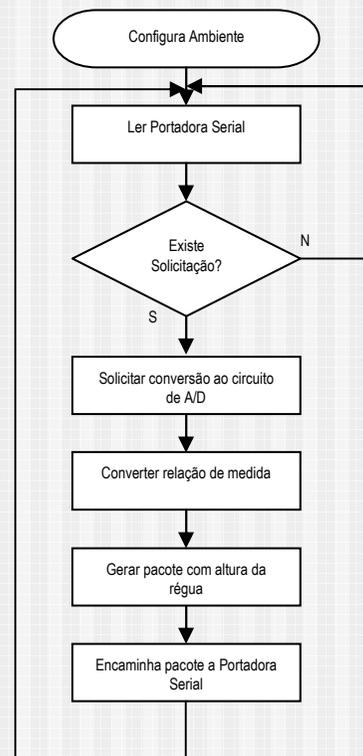
ESPECIFICAÇÃO

- Módulo de Coleta



ESPECIFICAÇÃO

■ Fluxograma



IMPLEMENTAÇÃO

```
BAUD_NO_INVERTED CON 396      'Taxa de Transferencia via Serial NON INVERTED
CR_SIN             CON "*"     'Caractere de inicio de pacote (START)
CR_END            CON "#"     'Caractere de termino de pacote (END)

' Define parametros do comando ADCIN
define    ADC_BITS    8          ' Numero de BITS de resulta
define    ADC_CLOCK   3          ' Fonte de clock (3=rc)
define    ADC_SAMPLEUS 50       ' Tempo de Sampling (em uS)

'Variaveis de controle do dado do nivel
VL_LTR  var word      'Valor da Leitura
VL_CVT  var word      'Valor convertido

TRISA=%00000001
TRISB=%00010000
START:
    IF PORTB.4 THEN          'Verifica solicitacao de Leitura
        'Le a porta do AD
        TRISA = %11111111    ' Set PORTA como entrada
        ADCON1 = 0           ' Set PORTA analogica
        adcin 0, VL_LTR      ' Le porda analogical 0 para VL_LTR

        'Monta pacote para Devolucao
        SEROUT2 PORTB.5,BAUD,[CR_SIN,VL_LTR,CR_END]
    ENDIF
    GOTO START
```

IMPLEMENTAÇÃO

- Banco de dados Borland Interbase.
- Ambiente de programação Borland Delphi;
- Linguagem Object Pascal;
- Linguagem PIC Basic;

RESULTADOS

- **Garantiu-se:**
 - Comunicação multiplexada a 2400 bps através do uso de duas frequências em FM (315 MHz e 433.92 MHz)
 - Garantiu-se a possibilidade de inclusão de novas réguas
 - Uma boa garantia de que os valores lidos são verídicos

EXTENSÕES

- Evoluir LOG, melhorando controle de erros e sua rastreabilidade;
- Criar funções para cálculos de intensidade, velocidade, profundidade e a frequência dos períodos de retorno de catastrofes;
- Implementar comparação temporal;
- Implementar demonstração gráfica em 3D, com recursos avançados de computação gráfica;
- Melhorar o fluxo de coleta;

DEMONSTRAÇÃO PRÁTICA
