

# *Especificação de Protocolos*

*FEUP/MRSC/AMSR*  
*MPR*

## *Bibliografia*

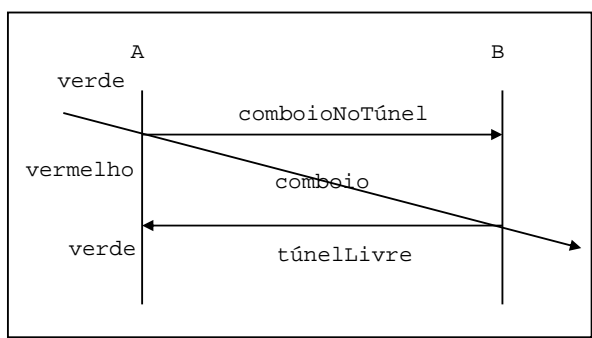
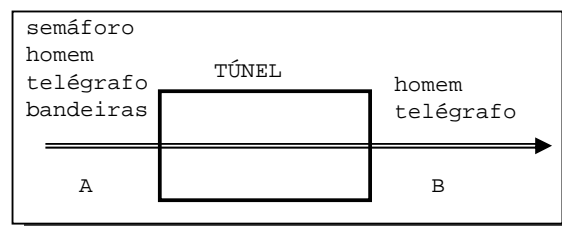
---

- » Gerard J. Holzmann, "Design and Validation of Computer Protocols", Prentice-Hall.
- » F. Belina, D. Hogrefe and A. Sarma, "SDL with applications from protocol specification", Prentice – Hall.
- » Telelogic, "Specification and Description Language (SDL)" tutorial.
- » D. Hogrefe, "OSI formal specification case study: the Inres protocol and service, revised", May 1992.

## Introdução

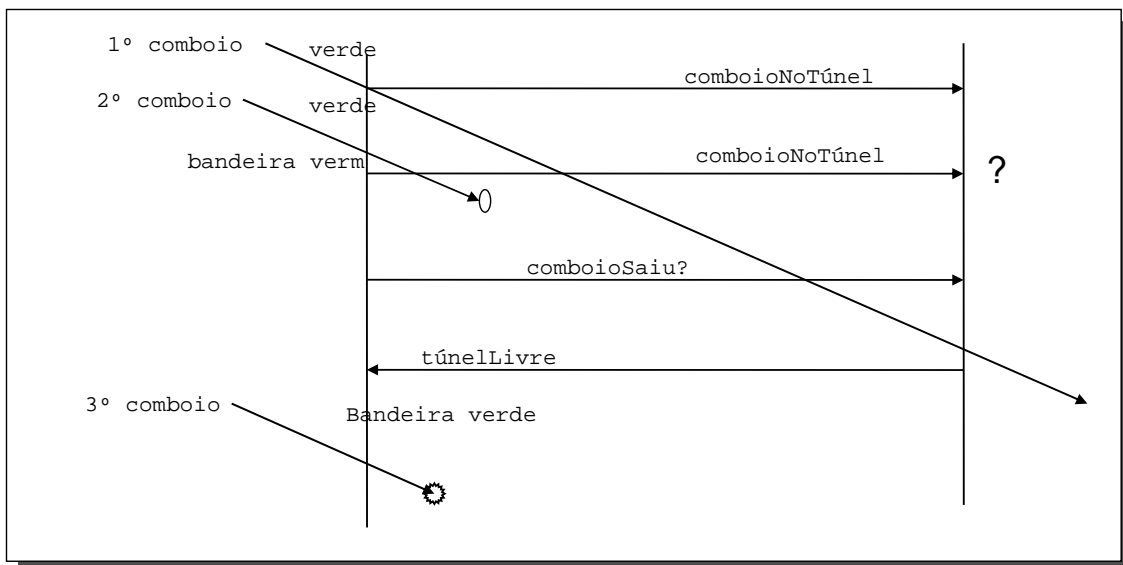
## Projecto de um Protocolo

- » Inglaterra, 1841
- » Túnel de 2 km



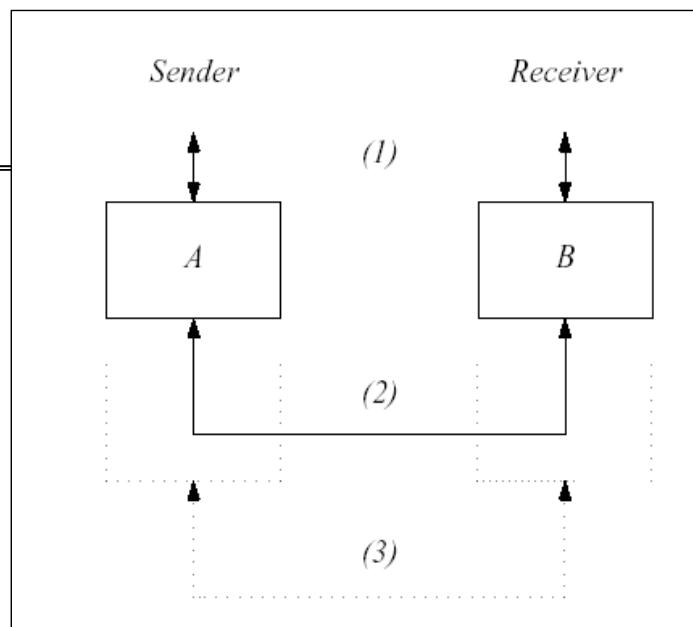
```
Telégrafo = {
  comboioNoTúnel,
  túnelLivre,
  comboioSaiu?
}
```

## Complexidade do Projecto



» 21 pessoas mortas, 176 feridas

## Protocolo de Comunicações



» Tendo

- A interface através da qual se recebe serviço → interface 2
- A definição abstracta do serviço a fornecer → interface 1

» Projecto de um protocolo consiste

- Transformar interface 2, passo a passo, até se conseguir fornecer serviço 1

## *Elementos Básicos de um Protocolo*

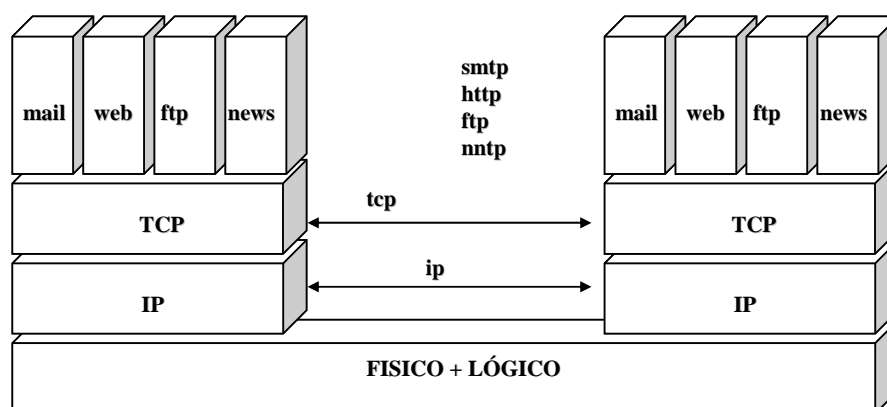
---

- » **Serviço** a fornecer
  - ◆ Formalização da interface superior
- » **Restrições**
  - ◆ Impostas pelo ambiente em que o protocolo é executado
  - ◆ Formalização da interface inferior
- » **Vocabulário**
  - ◆ mensagens usadas para implementar o protocolo
- » **Codificação** das mensagens do vocabulário
  - ◆ Sintaxe / formato de cada mensagem
- » **Procedimentos** que regulam a troca de mensagens
  - ◆ Gramática, regras de funcionamento

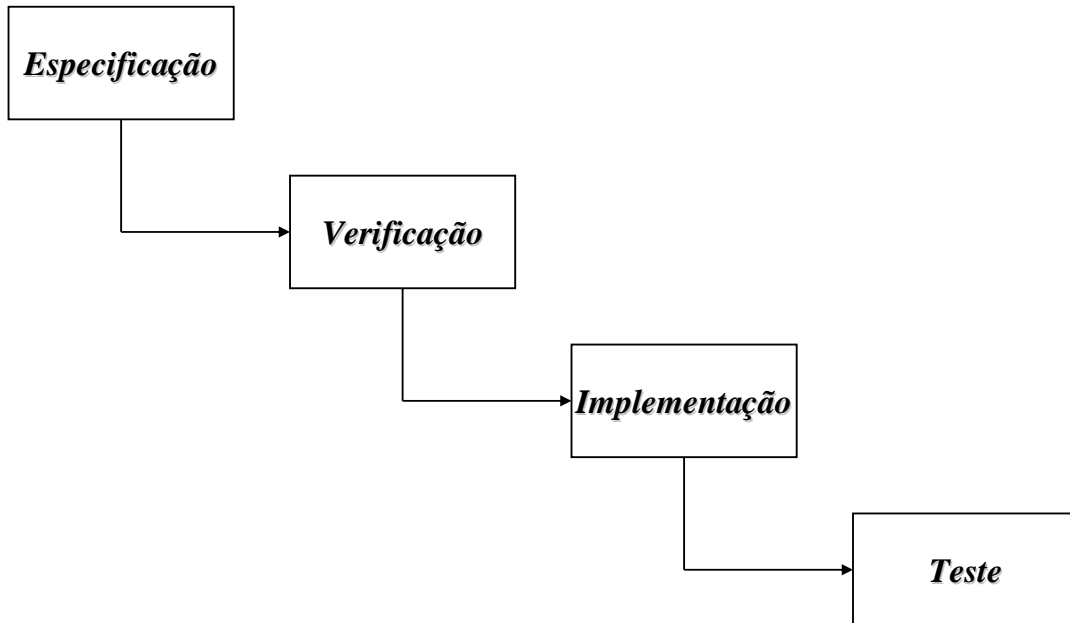
## *Pilhas de Protocolos*

---

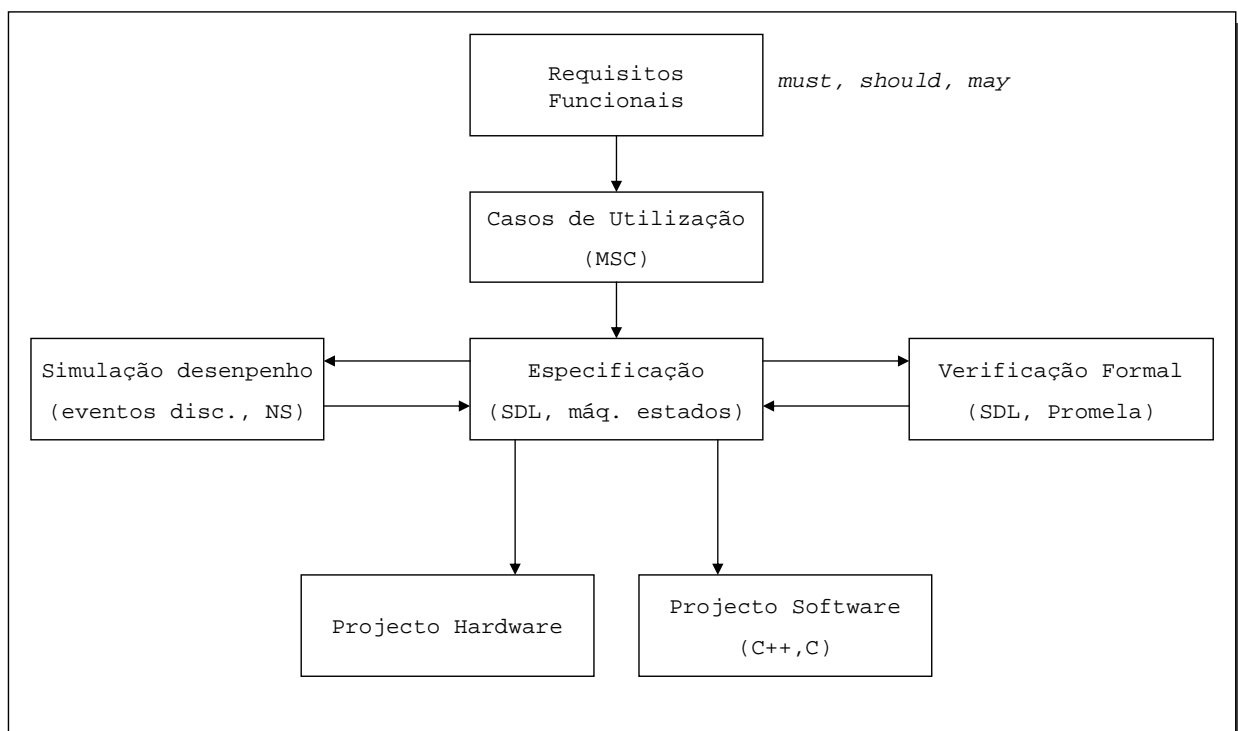
- » Exemplo, TCP/IP



## *Processo de Desenvolvimento de Um Protocolo*



## *Aproximação Possível*



---

# *Especificação de Protocolos*

## *SDL*

*(Specification and Description Language)*

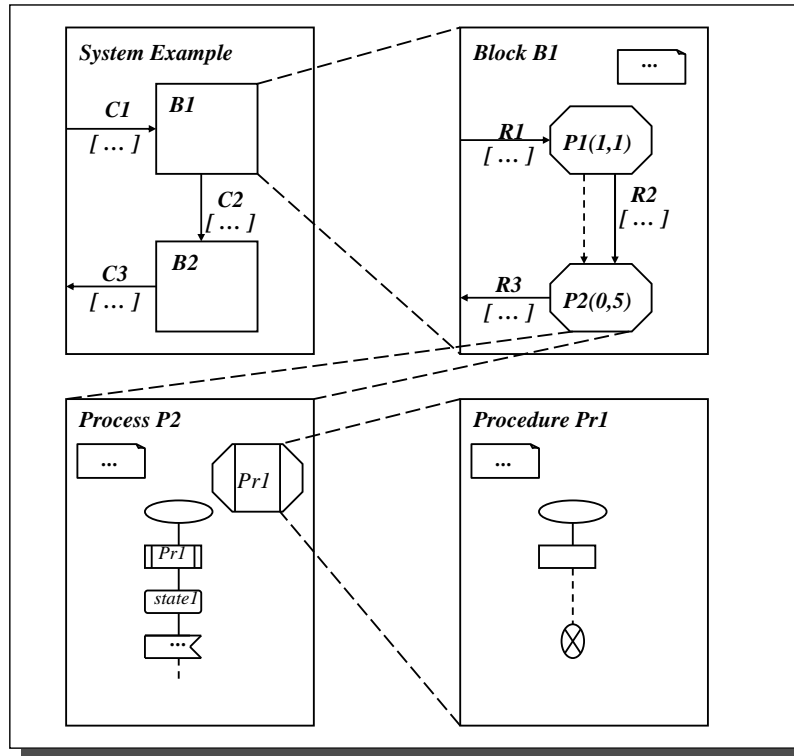
---

## *Linguagem de Especificação de Protocolos*

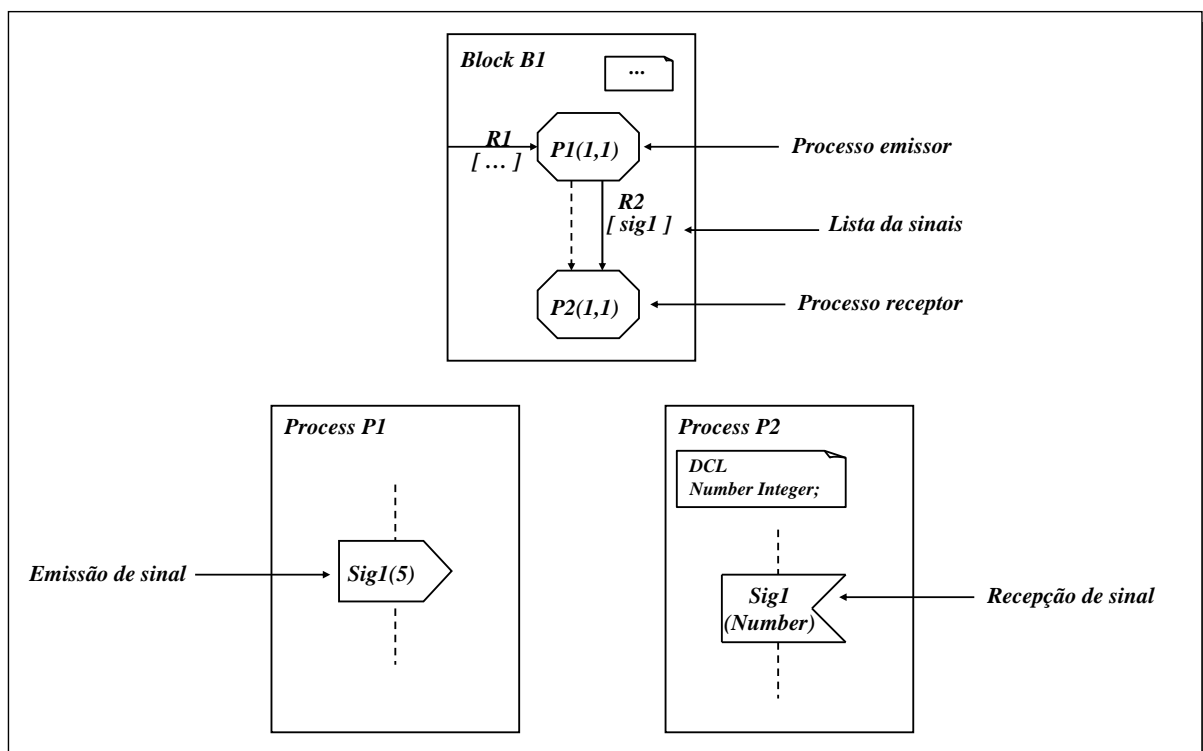
---

- ◆ *SDL - Specification and Description Language*
  - especificação de estrutura e funcionamento de sistemas
  - formal (estados, transições) → *comparação* de sistemas
  - representação gráfica e textual
  - suportada por CADs
  - normalizada por ITU (*Institute of Telecommunications Union*)
- ◆ *Vantagens*
  - Especificação, simulação e verificação de propriedades dos sistemas
  - geração automática de código (C, C++)
  - geração automática de testes para verificação de conformidade (TTCN)
- ◆ *Domínios de aplicação*
  - Telecomunicações (protocolos, sistemas)
  - Sistemas de tempo real

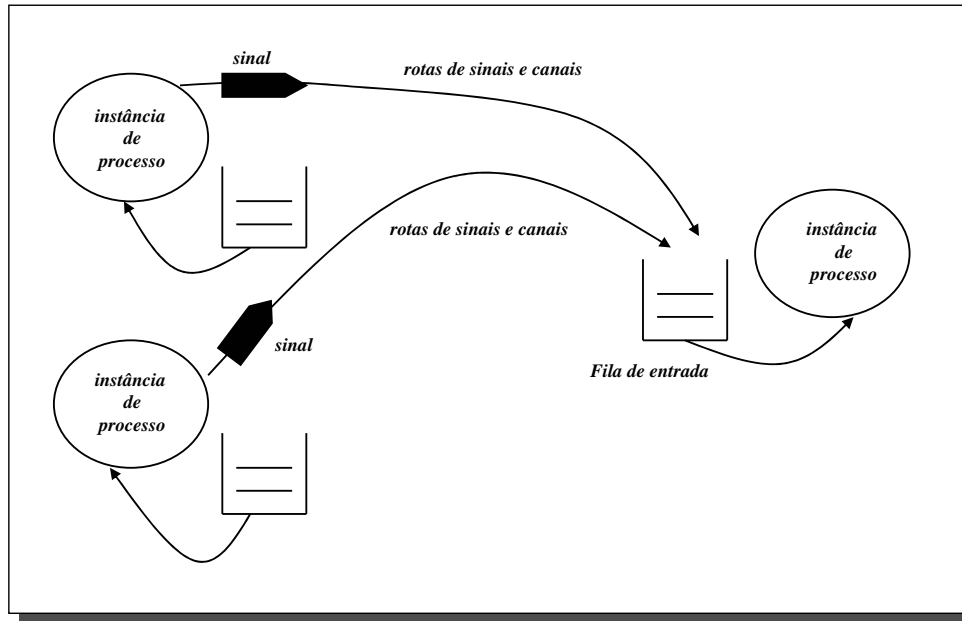
# Estrutura – Sistema, Bloco, Processo



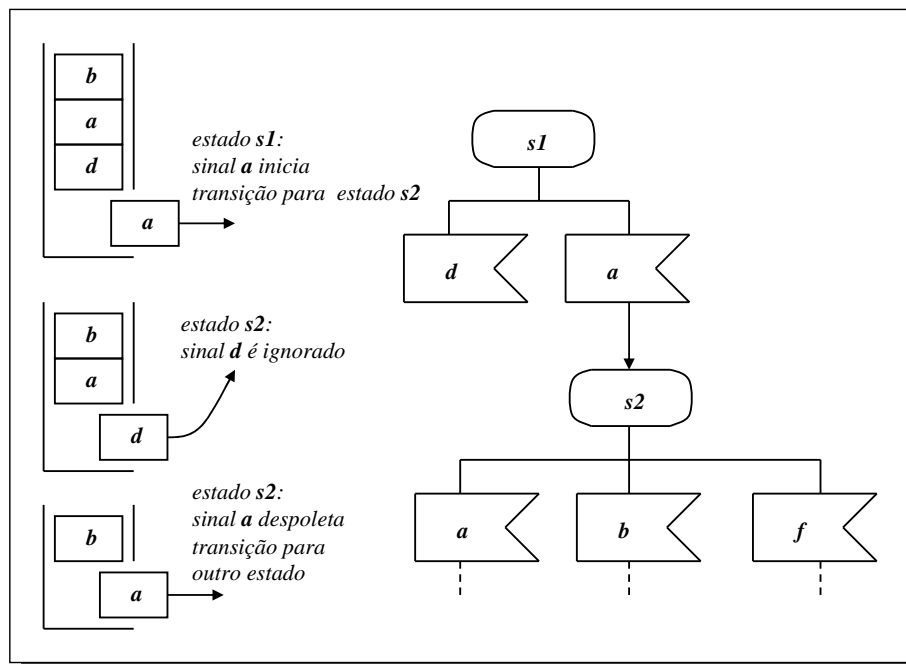
# Comunicação por Sinais



# Filas de Sinais

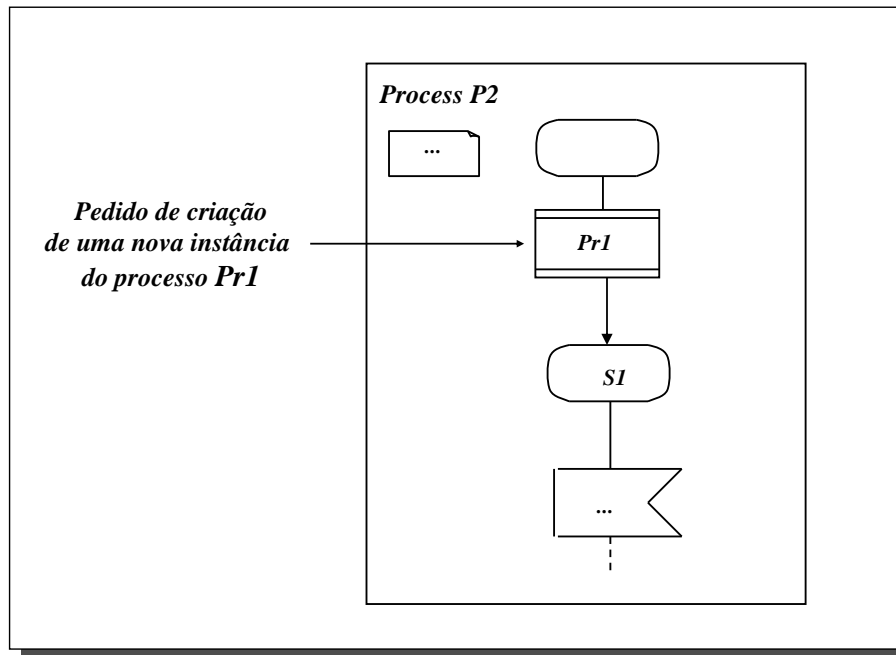


# Consumo de Sinais num Processo





## Criação Dinâmica de Instância de Processo



## Tipos de Dados

- » Tipos de dados predefinidos
  - ◆ Boolean, Character, Charstring
  - ◆ Integer, Natural, Real
  - ◆ Pid
  - ◆ Duration, Time
- » Definição de tipos novos e tipos sinónimos de dados
  - ◆ Newtype
  - ◆ Syntype
- » Cada instância de processo
  - ◆ tem endereço único no sistema
  - ◆ contêm 4 variáveis tipo *Pid*
    - *self* - endereço próprio
    - *sender* - endereço do processo que enviou o último sinal consumido
    - *offspring* - endereço do último processo criado
    - *parent* - endereço do processo pai

## *INRES - Especificação*

---

### » INRES - INItiator RESponder

- Protocolo *não real*
- Usado no estudo de testes de conformidade de protocolos
- Caso simples de um protocolo (ex. Transporte, nível 4)

### » Bom exemplo

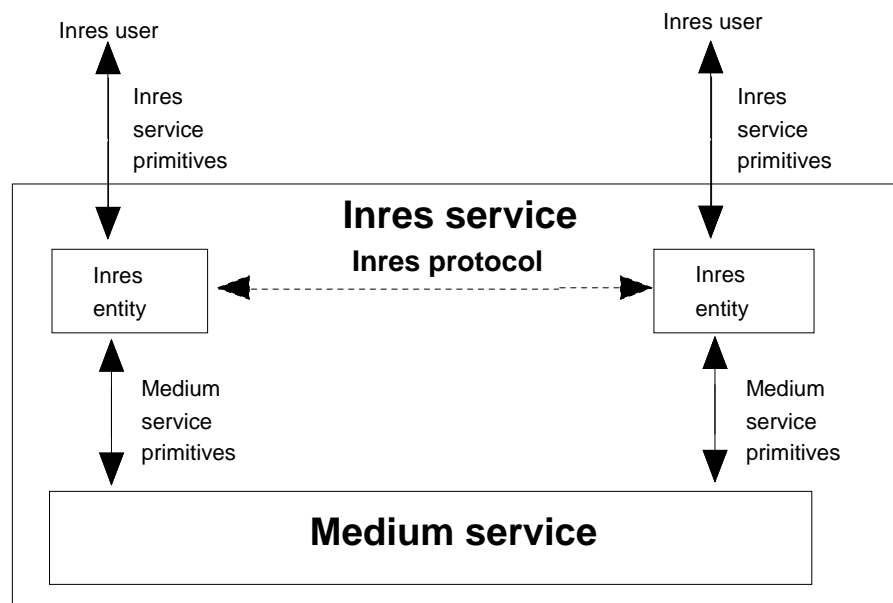
- Simples. Pequeno.
- Contêm aspectos essenciais dos protocolos de comunicação

### » 2 serviços, 1 protocolo

- *serviço INRES*: serviço de transporte orientado às ligações
- *serviço Medium*: transmissão não fiável de PDUs
- *protocolo INRES*: regras de comunicação entre o Initiator e o Responder

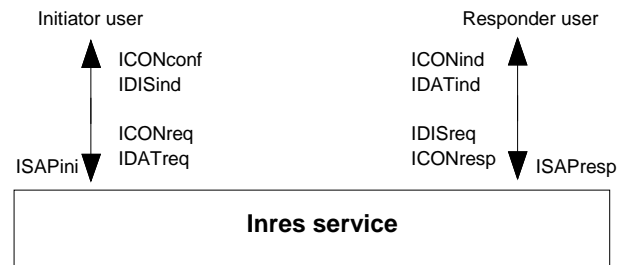
## *Arquitectura do Serviço INRES*

---



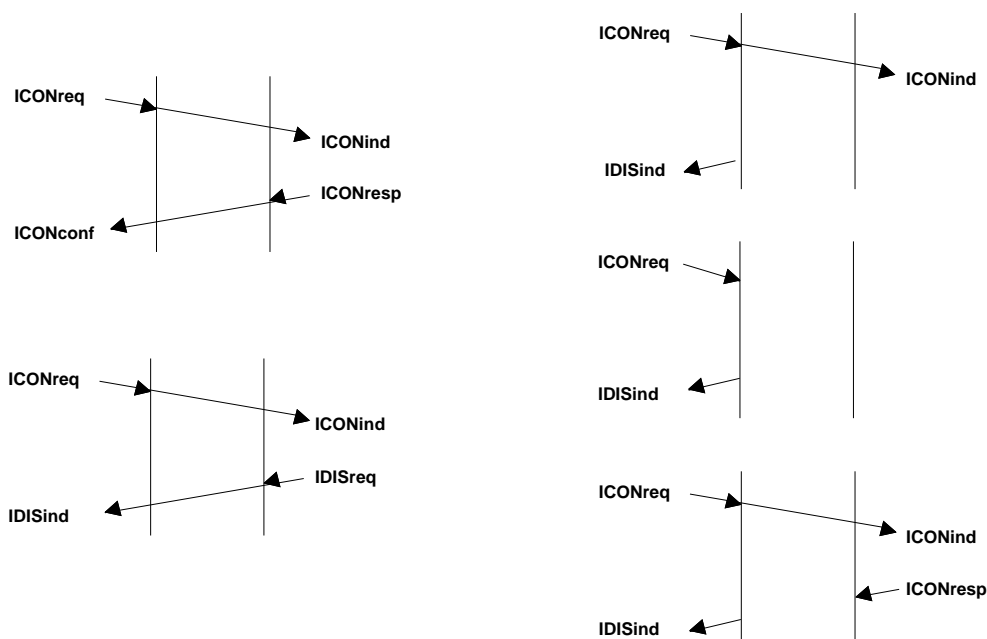
## Serviço INRES

- ◆ Orientado às ligações
  - » Estabelecer ligação, transferir dados, terminar ligação
- ◆ 2 pontos de acesso de serviço (SAP)
  - » *ISAPini*, *ISAPresp*
  - » Serviço não simétrico
  - » Chamada
    - Iniciada em *ISAPini*
    - Com dados de *ISAPini* → *ISAPresp*
    - Terminada em *ISAPresp*

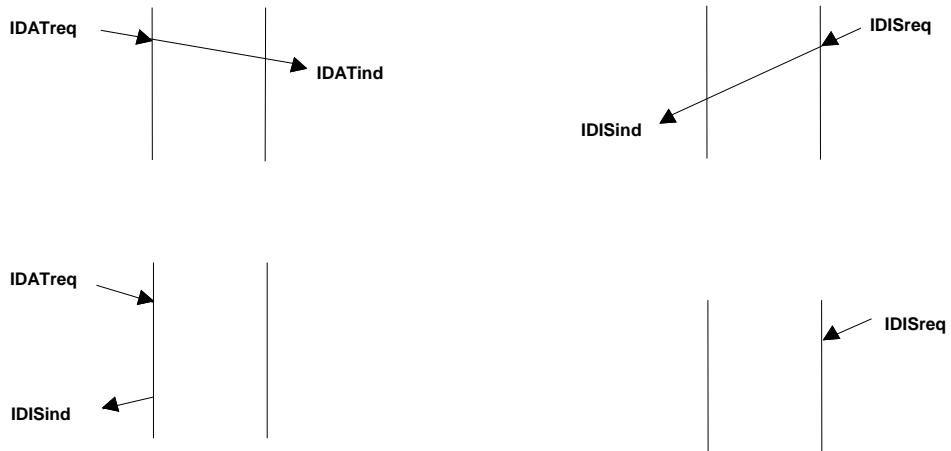


- ◆ Primitivas de serviço
  - » *ICONreq*: pedido de ligação
  - » *ICONind*: indicação de ligação
  - » *ICONresp*: resposta de pedido de ligação
  - » *ICONconf*: confirmação aceitação
  - » *IDATreq*(ISDU): pedido transfer. dados (ISDU)
  - » *IDATind*(ISDU): indicação de dados
  - » *IDISreq*: pedido de terminação de ligação
  - » *IDISind*: indicação de terminação de ligação

## Serviço INRES – Estabelecimento de Ligação



# Serviço INRES – Transferência de Dados e Terminação de Ligação



# Serviço Medium

◆ Transporte *não fiável* de dados

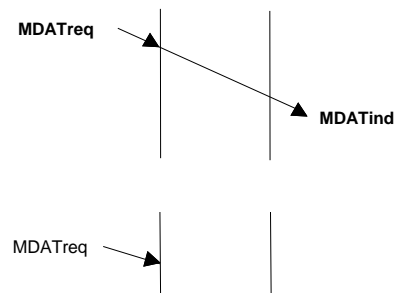
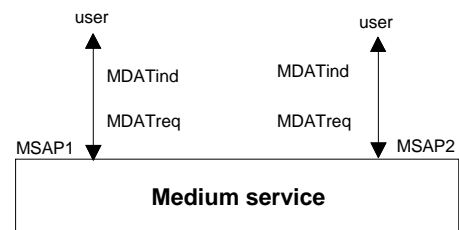
- Dados podem ser perdidos
- Dados não são corrompidos
- Dados não são duplicados

◆ 2 pontos de acesso de serviço (SAP)

- *MSAPI*, *MSAP2*
- Serviço simétrico

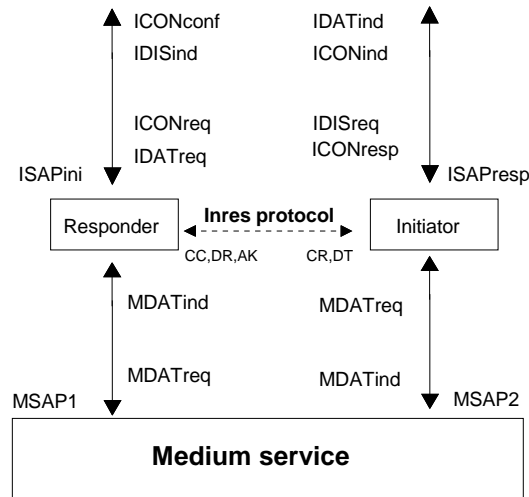
◆ Primitivas de serviço

- MDATreq(MSDU): pedido transfer. dados (MSDU)
- MDATind(MSDU): indicação de dados



## Protocolo INRES – PDUs, Semântica

<i>PDU</i>	<i>Significado</i>	<i>Parâmetros</i>	<i>Primitivas associadas</i>
CR	connection establishment	none	ICONreq, ICONind
CC	connection confirmation	none	ICONresp, ICONconf
DT	data transfer	sequence number, ISDU	IDATreq, IDATind
AK	acknowledgement	sequence number	-
DR	disconnection	none	IDISreq, IDISind



## Procedimentos - Estabelecimento De Ligação

- » Iniciada por utilizador do **Initiator** com **ICONreq**
- » **Initiator** envia **CR** para **Responder**
- » **Initiator** pode receber do **Responder** : **CC**, **DR** ou **nada**
  - **CC**: **Initiator** envia **ICONconf** para seu utilizador → **Transferência De Dados**
  - **DR**: **Initiator** envia **IDISind** para seu utilizador → **Terminação De Ligação**
  - Nada:
    - ◆ fim de 5s, **Initiator** envia **CR** de novo
    - ◆ Ao fim de 4 tentativas → **Terminação De Ligação**

## Procedimentos – TransferênciaDeDados

---

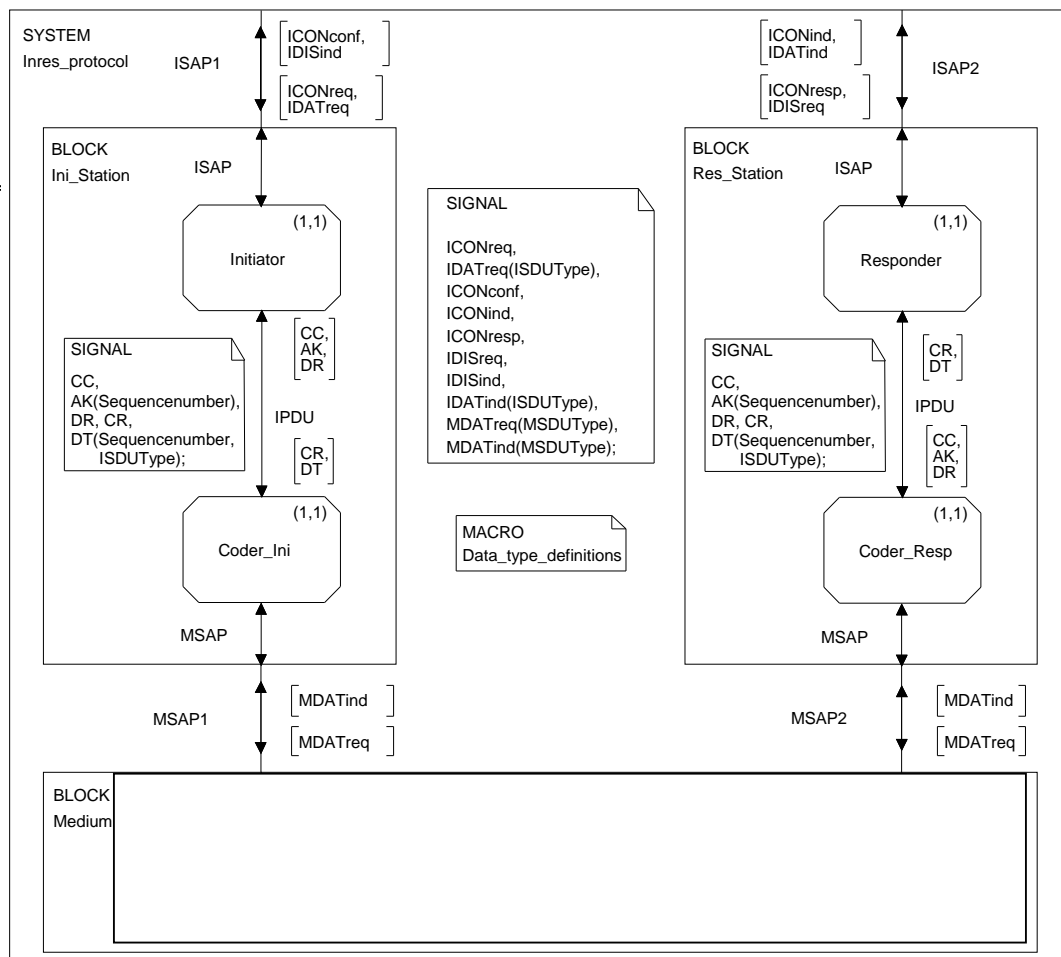
- » Utilizador do **Initiator** envia **IDATreq(ISDU)**
  - ◆ **ISDU** contém informação da **camada superior** ( PDU + info controlo)
  - ◆ **Initiator** envia **DT** para **Responder**
  - ◆ PDU da camada superior transportado transparentemente
  - ◆ **Initiator** espera até **5s** por confirmação **AK**
  - ◆ Se não chegar **AK**, retransmite DT.
  - ◆ Ao fim de 4 tentativas → **TerminaçãoDeLigação**
  
- » **DT, AK** contêm número sequência. 2 valores possíveis: **0,1**
  - ◆ **Initiator** começa com valor **1**
  - ◆ espera confirmação com mesmo valor: **DT( 1, ... ) → AK( 1 )**
  - ◆ Se recebe confirmação
    - com mesmo valor → transmite novos dados, **DT( 0, ... )**
    - com valor diferente (erro) retransmite dados, **DT( 1, ... )**

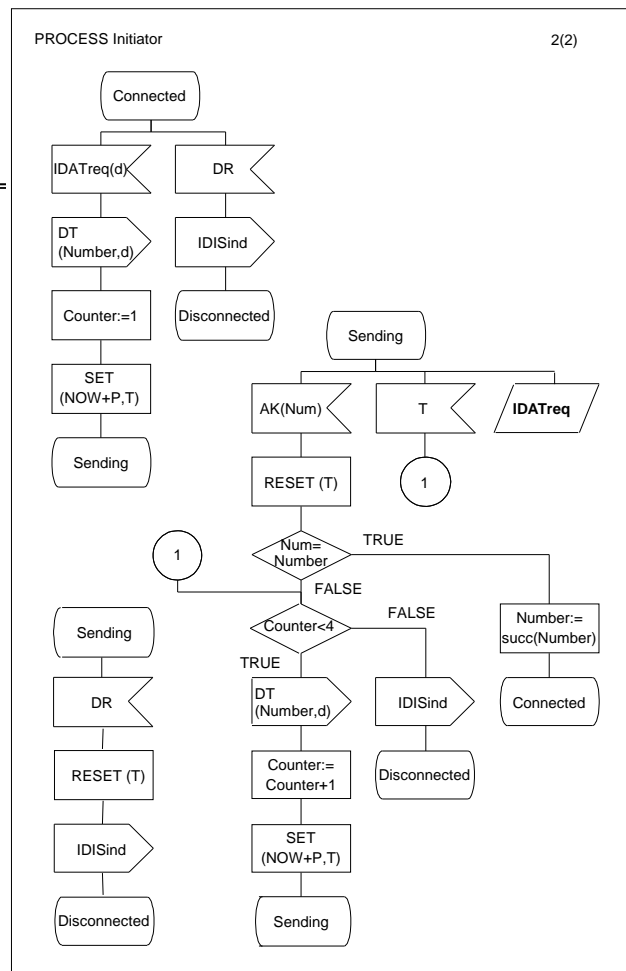
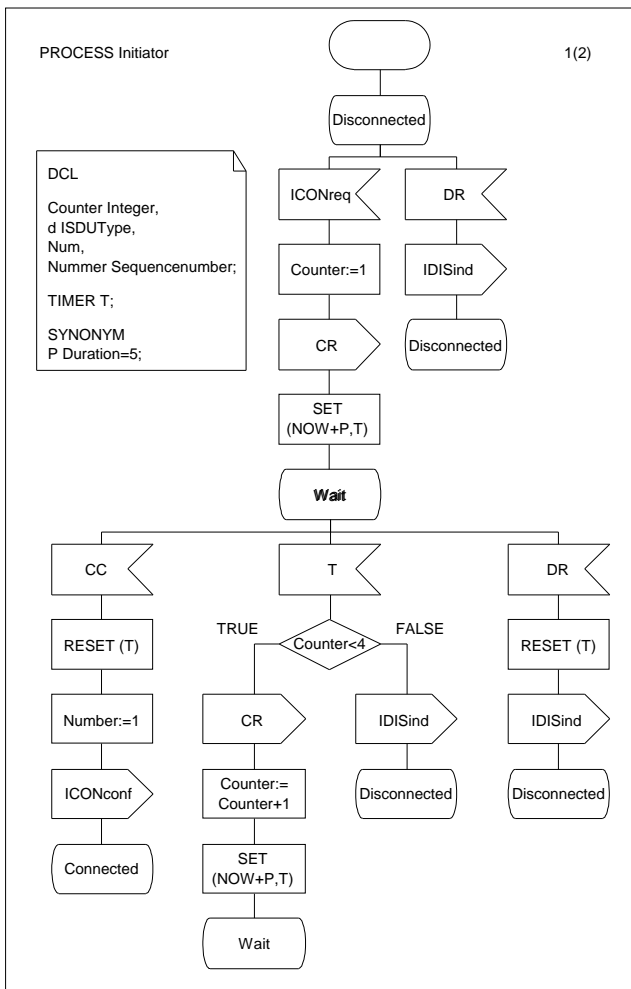
## Procedimentos – TerminaçãoDeLigação

---

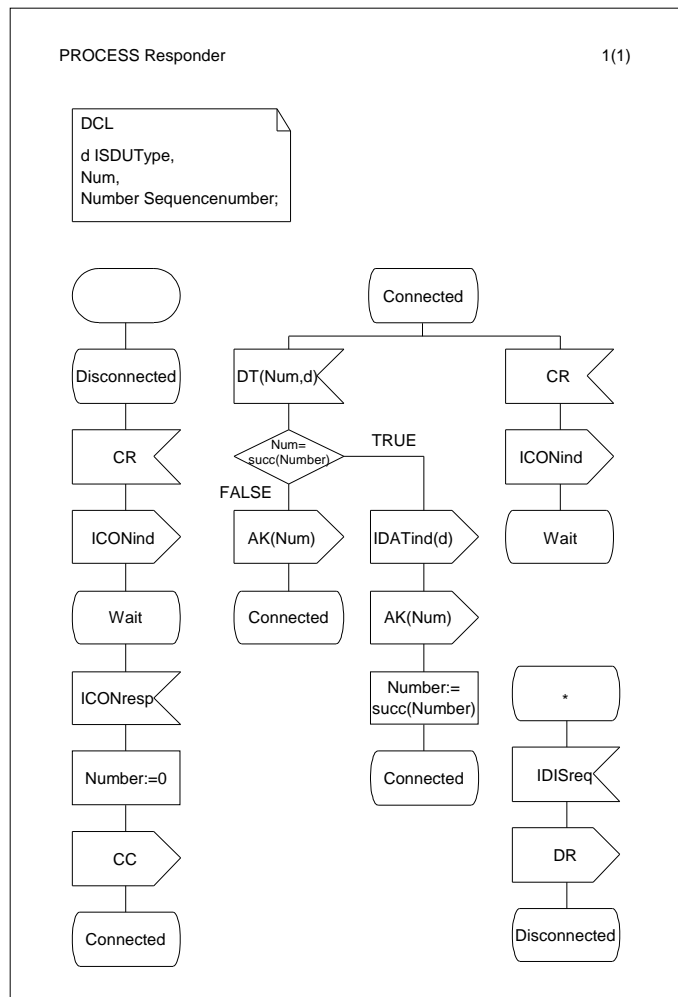
- » **IDISreq** enviado por utilizador de **Responder**
  - **DR** enviado por **Responder** para **Initiator** → **EstabelecimentoDeLigação**
  
- » No **Initiator**
  - **DR** recebido
  - **IDISind** enviado pelo **Initiator** para o seu utilizador → **EstabelecimentoDeLigação**

# INRES em SDL



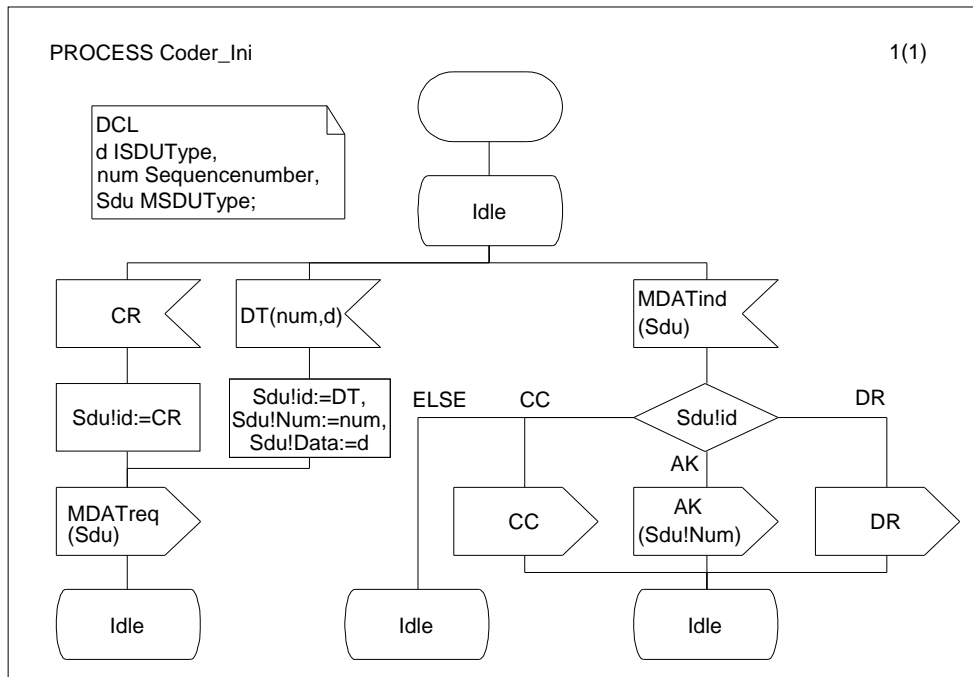


# Responder





# Coder\_ini



# Coder\_Resp

