



# Protótipo de um mundo virtual distribuído utilizando DIS-Java-VRML

Edison Weise

Prof. Dalton dos Reis



# Roteiro da Apresentação

- Introdução
- Ambientes Virtuais Distribuídos - AVD
- Linguagem VRML
- Protocolo DIS
- Tecnologia DIS-Java-VRML
- Especificação e Implementação
- Conclusão



# Introdução

Objetivo principal – implementar um protótipo de um mundo virtual distribuído e não imersivo sobre uma rede local, utilizando a tecnologia DIS-Java-VRML

Os objetivos específicos do trabalho foram:

- representar objetos por primitivas gráficas simples
- permitir a interação com o mundo virtual através de um avatar



# AVD: Conceitos

- Realidade Virtual

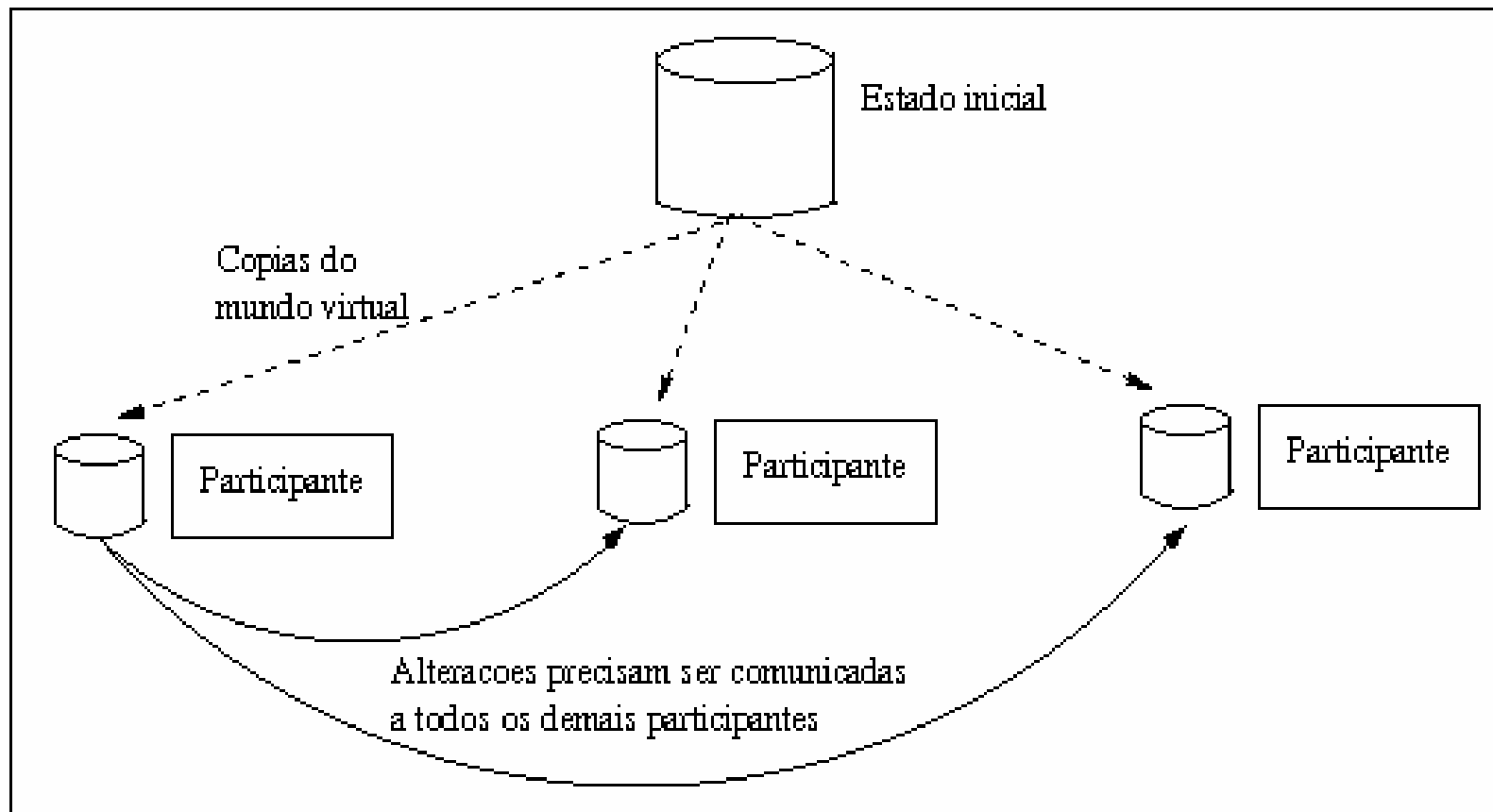
- Uma forma das pessoas visualizarem, manipularem e interagirem com computadores e dados extremamente complexos (Kirner, 2000)

- Ambientes imersivos e não imersivos

- O que são Ambientes Virtuais Distribuídos

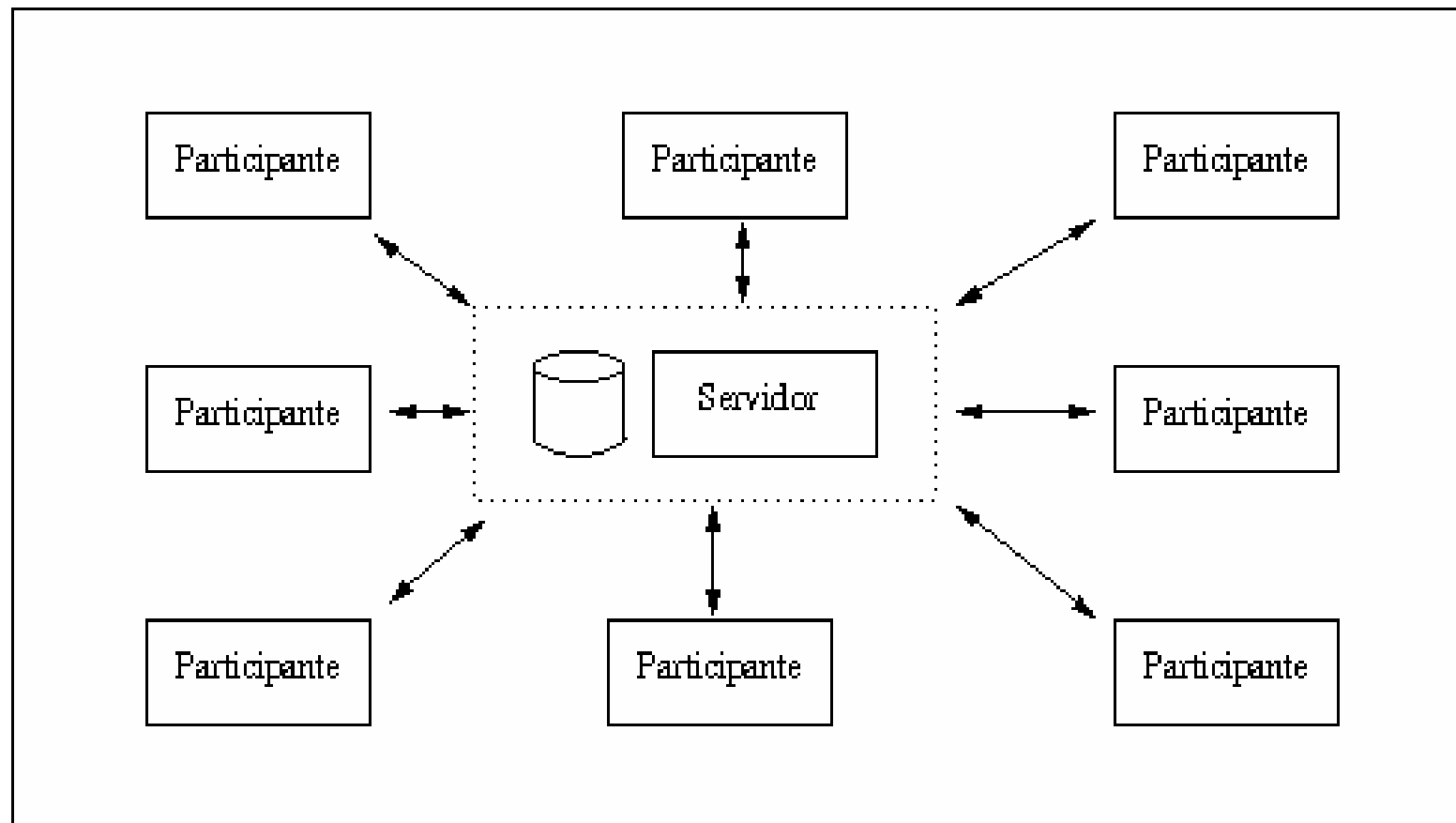
- Tipos de Ambientes Virtuais Distribuídos

# AVD: Tipos – Homogêneo Replicado



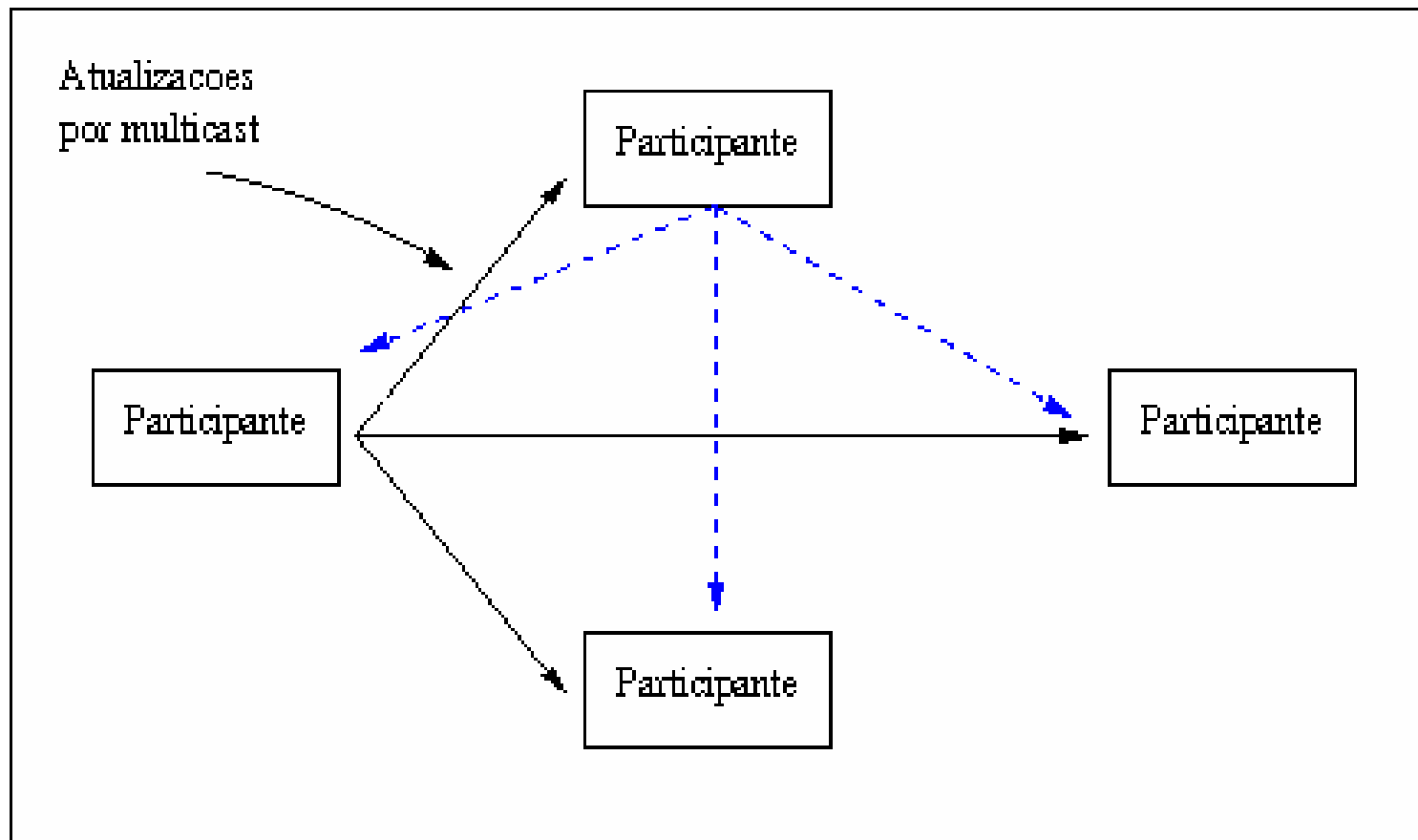
Fonte: Raposo, 2000

# AVD: Tipos – Centralizado



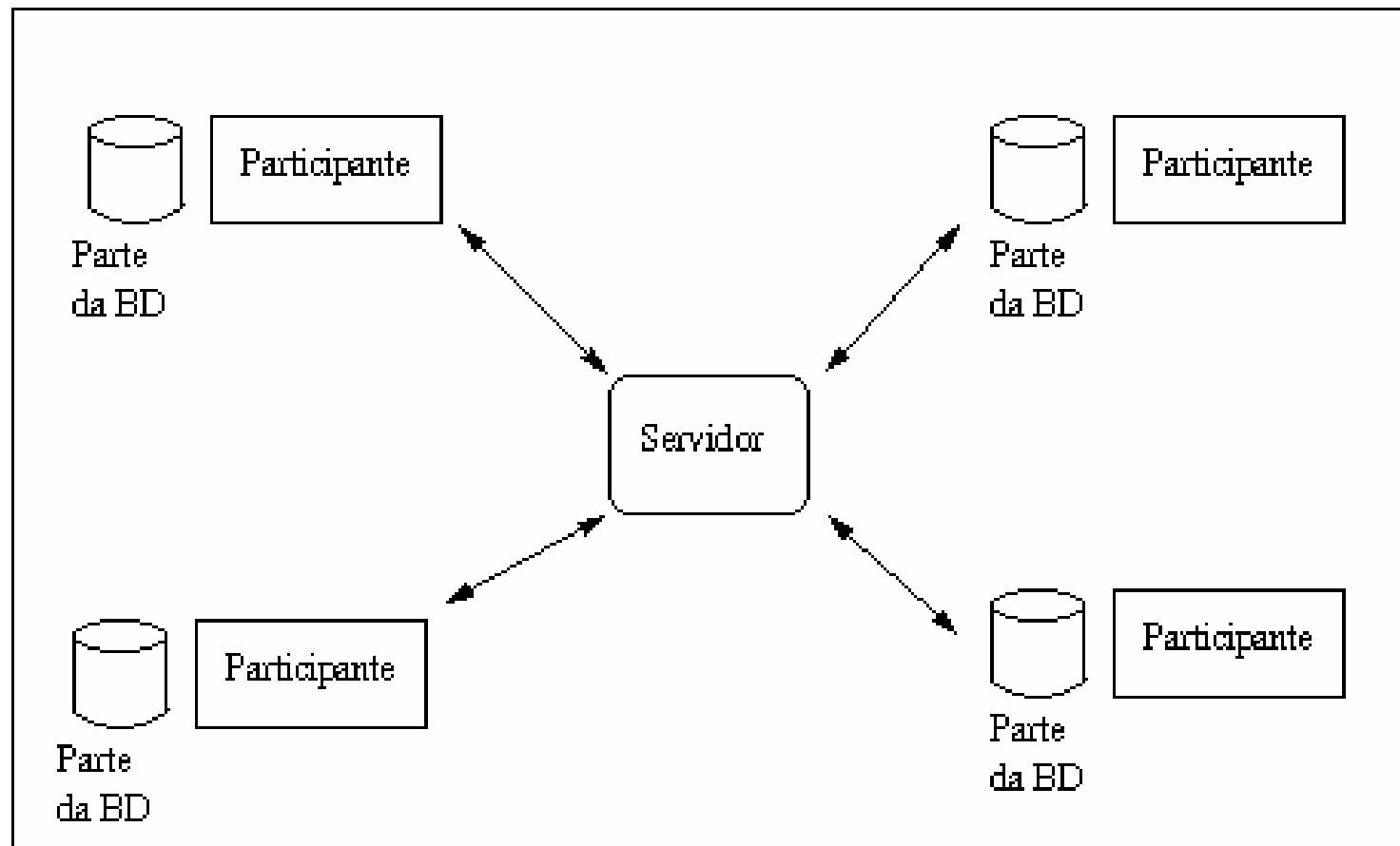
Fonte: Raposo, 2000

# AVD: Tipos – Distribuído Ponto a Ponto



Fonte: Raposo, 2000

# AVD: Tipos – Distribuído Cliente-Servidor



Fonte: Raposo, 2000






# VRML: Conceitos

- *Virtual Reality Modeling Language*
- Versão atual – VRML 2.0
- Padrão atual de especificação – VRML97
- Web3D Consortium – [www.web3d.org](http://www.web3d.org)



# VRML: Funções – Objeto

```
Shape {  
  geometry Box { # forma de uma caixa  
    size 2 2 5 # largura, altura, profundidade  
  }  
} # fim do shape
```



# VRML: Funções – Transformação

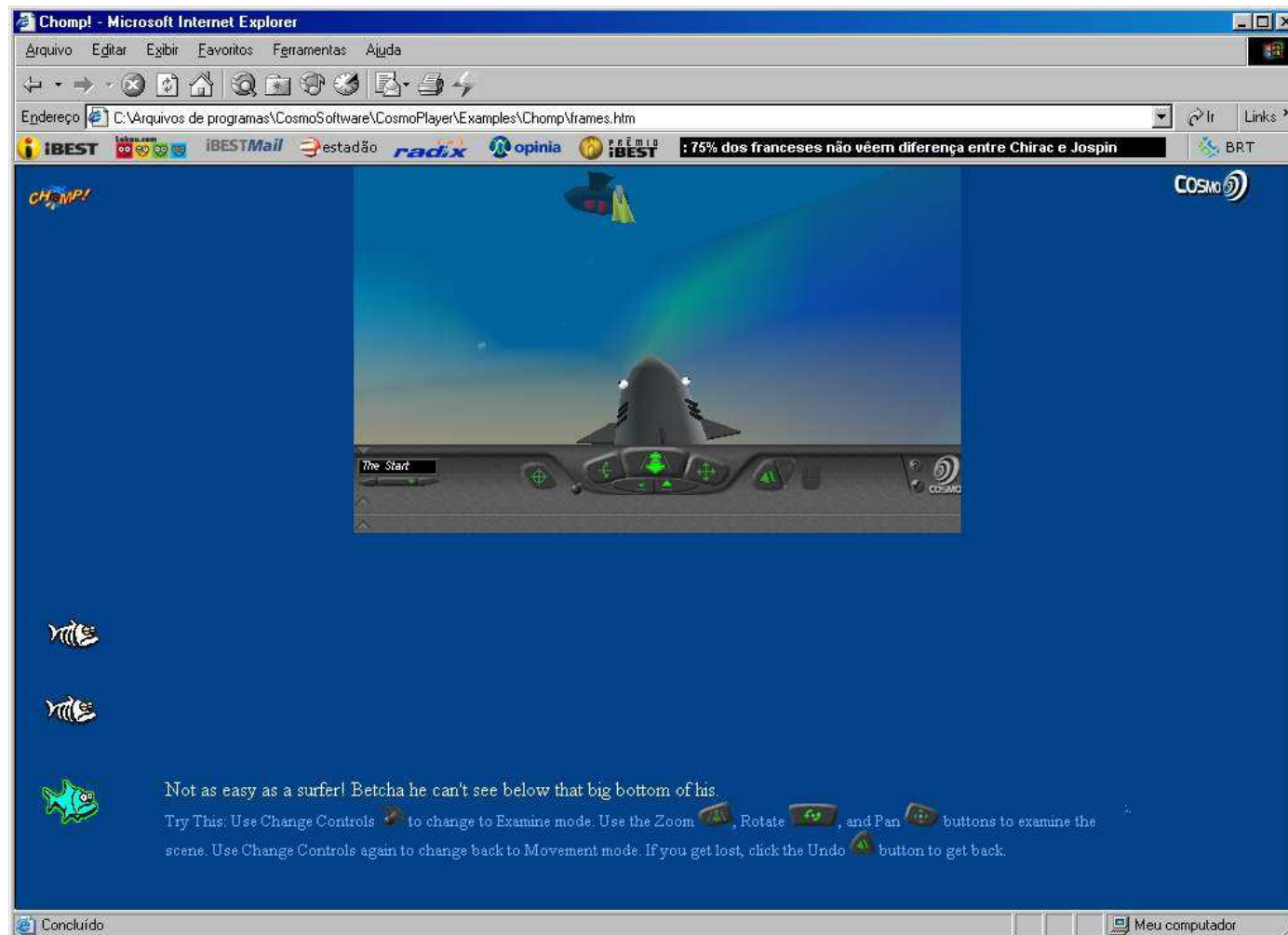
```
Transform {
  translation 8 0 0 # coordenadas x,y,z
  rotation 0 0 1 0.7 # eixo de rotação e
                    # ângulo (radianos)
  scale 1 1 2 # coordenadas x,y,z
  children [
    Shape {
      geometry Box {
        size 5 5 5 # larg,alt,profundidade
      }
    }
  ]
} # fim do shape
```



# VRML: Funções – Aparência

```
Shape {  
  appearance Appearance {# aparência do shape  
    material Material {  
      diffuseColor 1.0 1.0 0.0    # cor RGB  
    } # fim do material  
    texture ImageTexture {  
      url ["imagem.jpg"]  
    } # fim do texture  
  } # fim do appearance  
} # fim do shape
```

# VRML: Exemplo – Chomp





# DIS: Conceitos

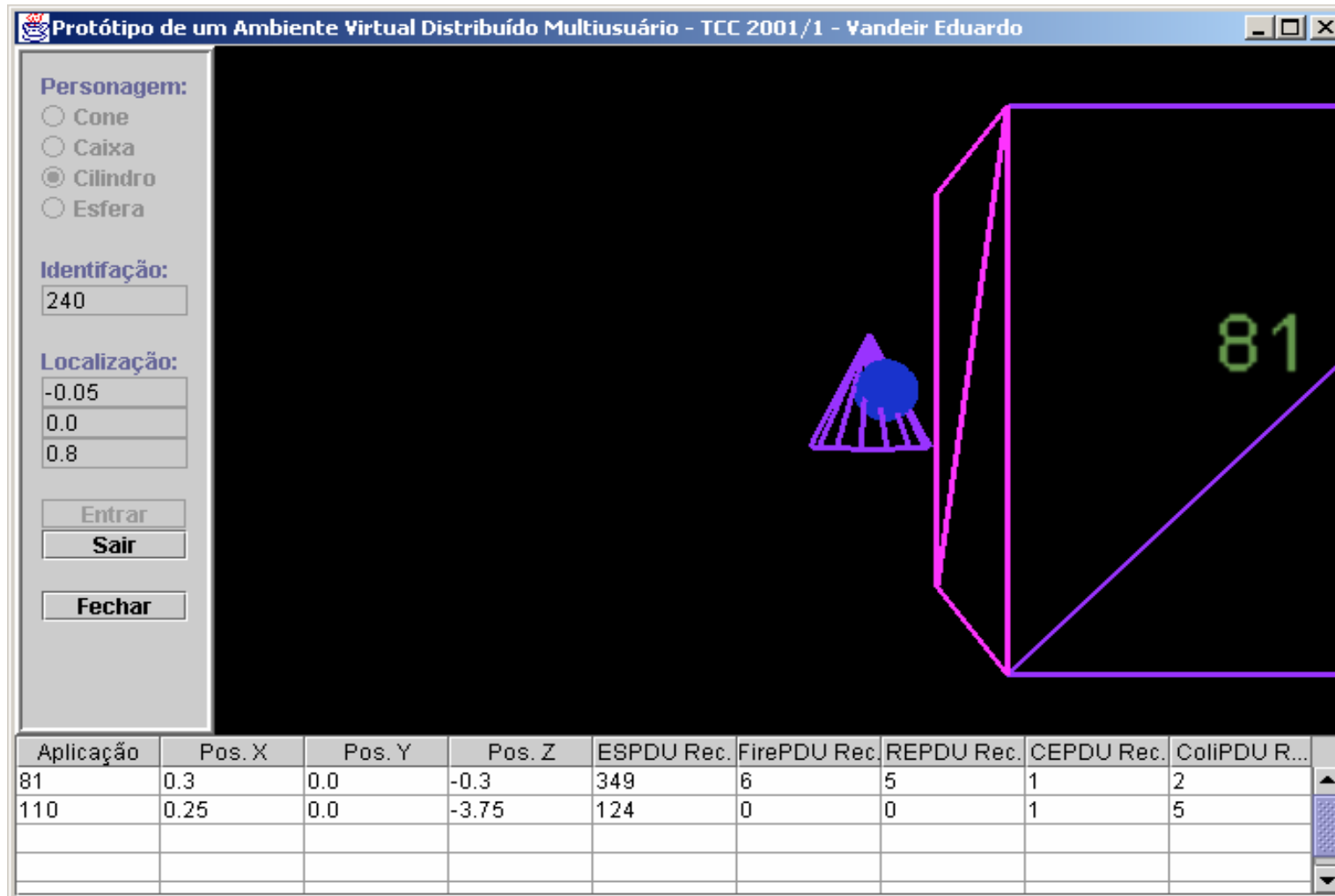
- *Distributed Interactive Simulation*
- Endereçamento *Broadcast*
- Protocol Data Unit
  - EntityStatePdu
  - CollisionPdu
  - DetonationPdu
  - FirePdu

# DIS: EntityStatePdu – Estrutura

Nome do Campo	Conteúdo/Assunto	Tamanho Campo (em bytes)
Cabeçalho do PDU	Versão protocolo, ID do exercício, tipo do PDU, família do protocolo, <i>timestamp</i> , tamanho	12
ID da entidade	<i>Site</i> , aplicação, entidade	6
ID da força		1
Número de parâmetros de articulação		1
Tipo de entidade	Espécie de entidade, domínio, país, categoria, subcategoria, dados específicos, extras	8
Tipo entidade alternativa	Espécie de entidade, domínio, país, categoria, subcategoria, dados específicos, extras	8
Velocidade linear	x, y, z	12
Localização	x, y, z	24
Orientação	Psi, theta, phi	12
Aparência		4
Parâmetros de <i>Dead Reckoning</i>	Algoritmo, outros parâmetros, aceleração linear e velocidade angular da entidade	40
Sinalizações da entidade	Grupo de caracteres, sinalizações	12
Habilidades		4
Parâmetros de articulação	Tipo de parâmetro, mudança, ID, valor do parâmetro	N * 16

Fonte: Adaptado de Singhal (1999)

# DIS-Java-VRML: Exemplo



Protótipo de um Ambiente Virtual Distribuído Multiusuário - TCC 2001/1 - Vandeir Eduardo

**Personagem:**

- Cone
- Caixa
- Cilindro
- Esfera

**Identificação:**

240

**Localização:**

-0.05

0.0

0.8

Entrar

Sair

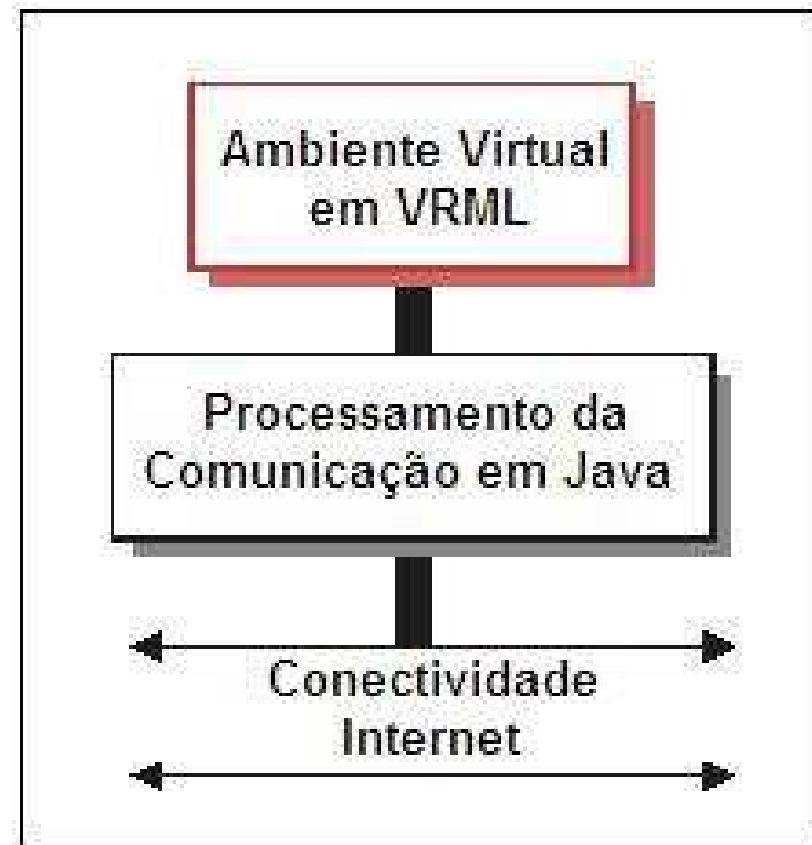
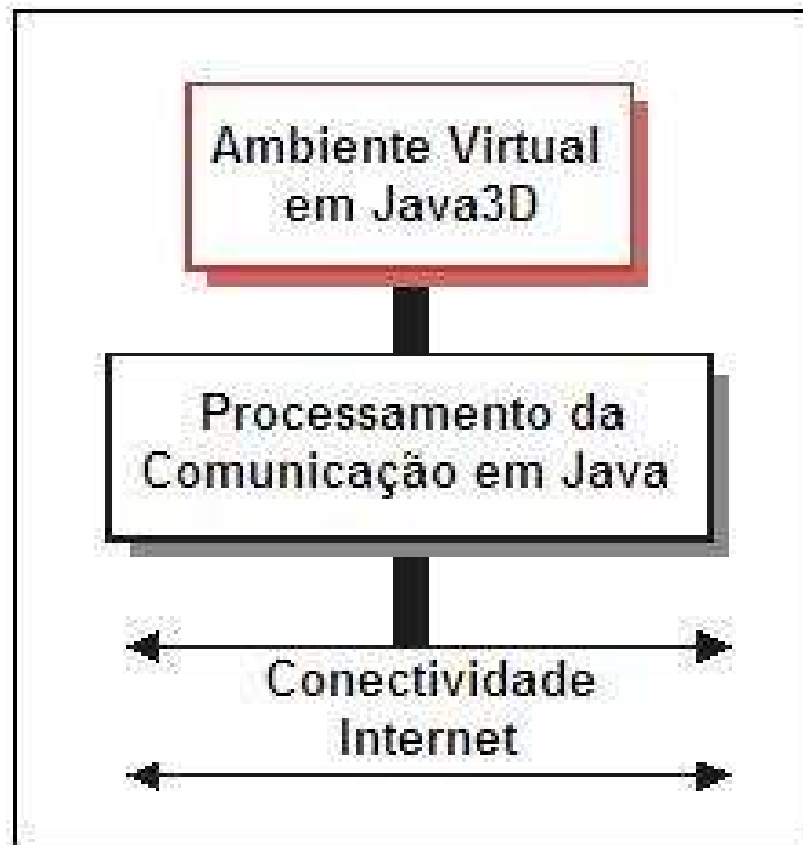
Fechar

Aplicação	Pos. X	Pos. Y	Pos. Z	ESPDU Rec.	FirePDU Rec.	REPDU Rec.	CEPDU Rec.	ColiPDU R...
81	0.3	0.0	-0.3	349	6	5	1	2
110	0.25	0.0	-3.75	124	0	0	1	5

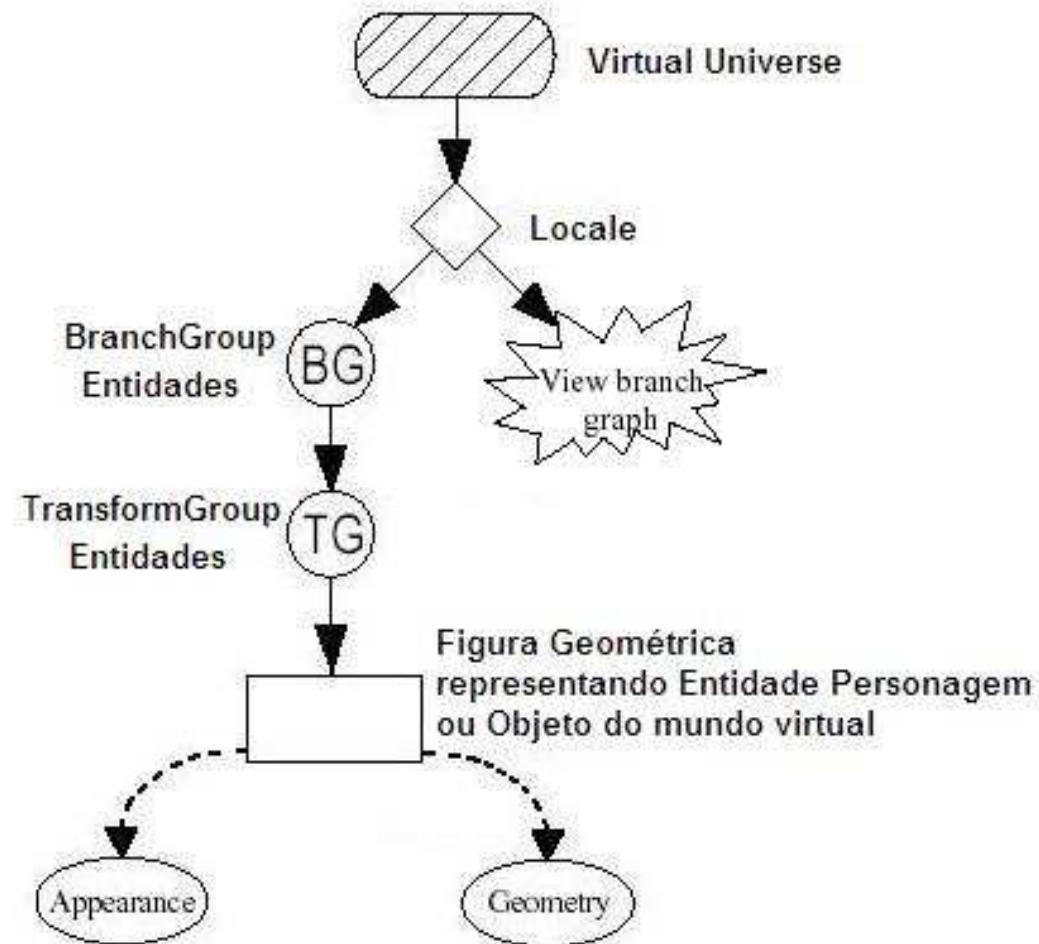
Fonte: Eduardo (2001)



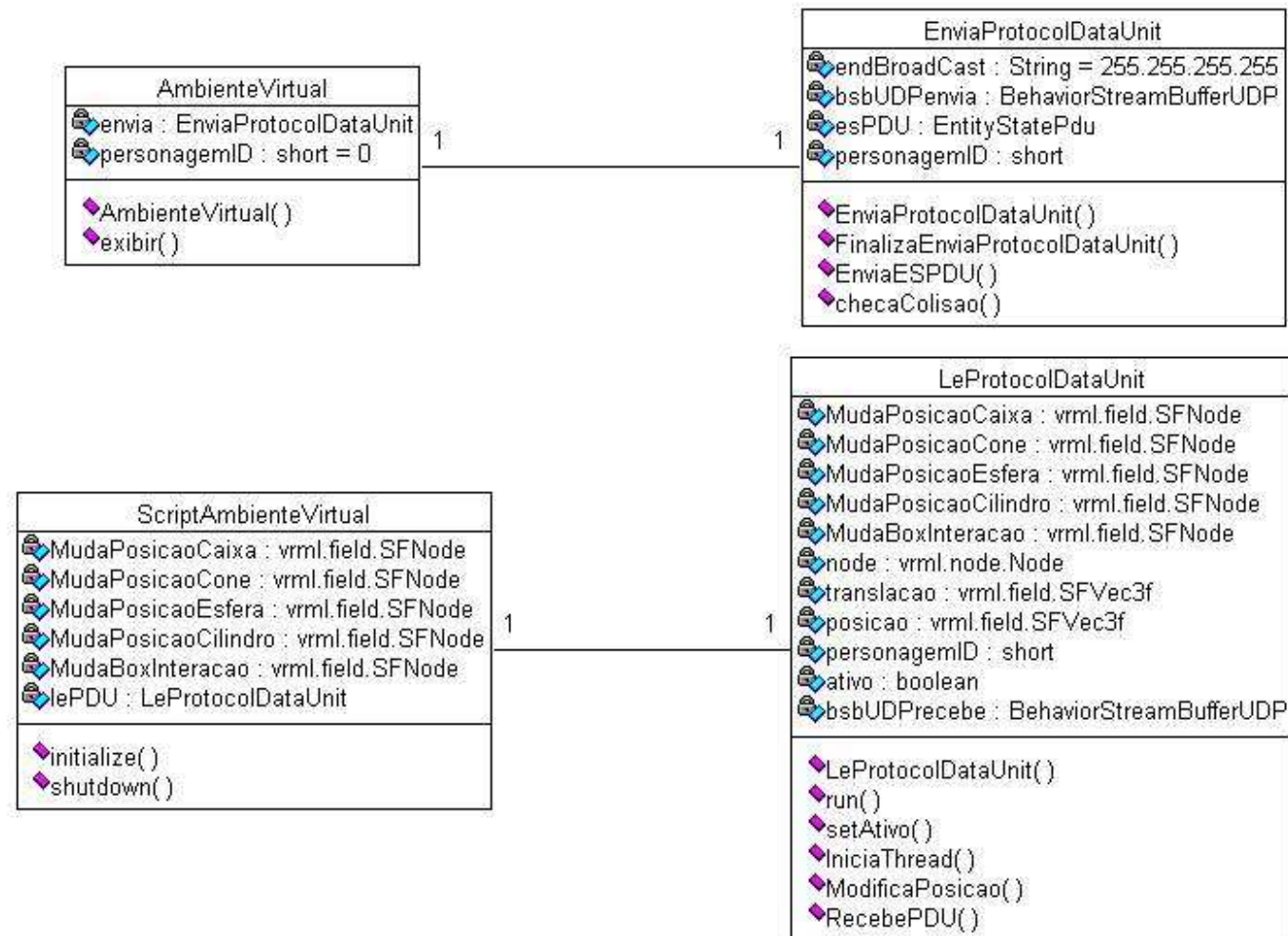
# DIS-Java-VRML: Exemplo



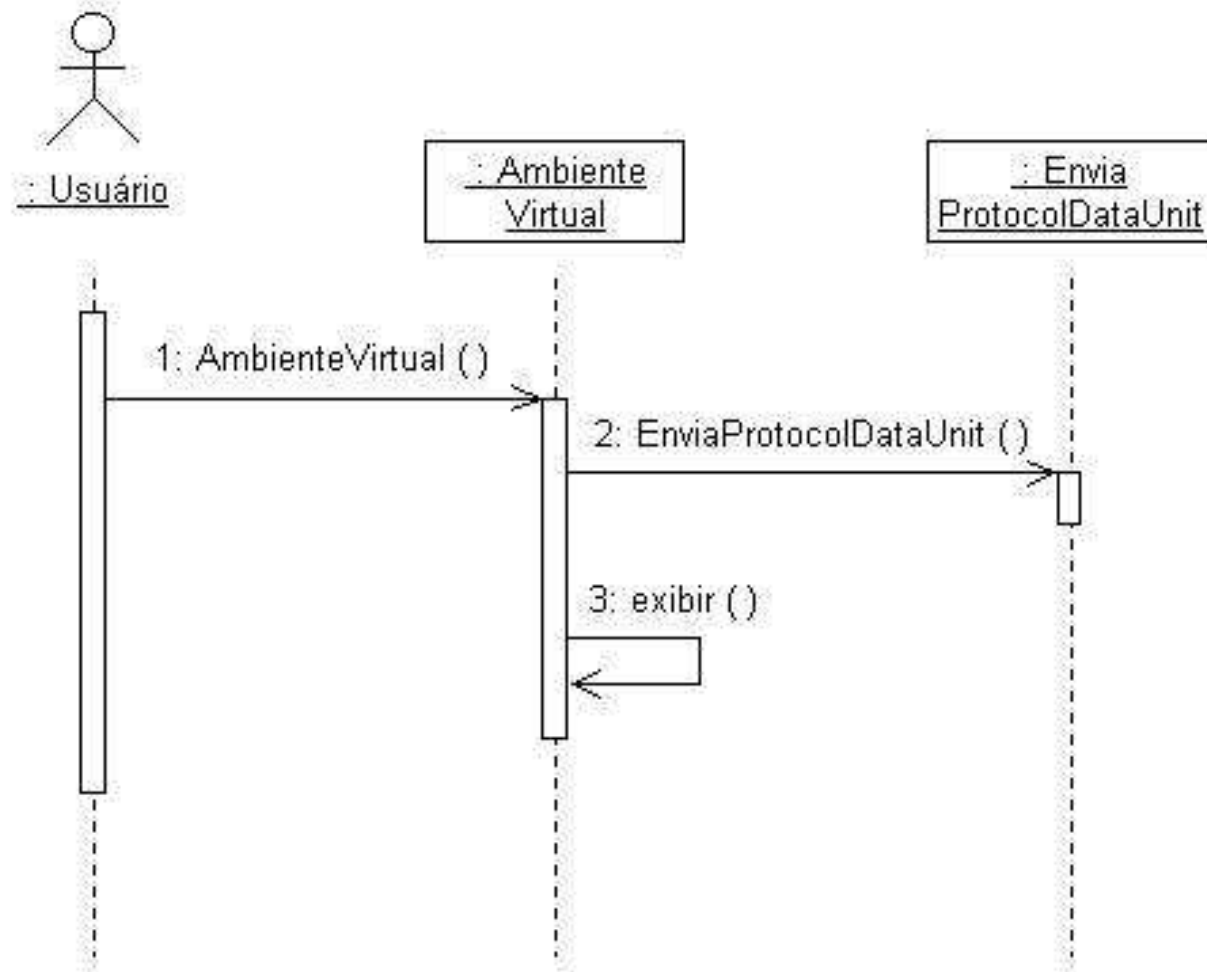
# Especificação: Grafo de Cena



# Especificação: Diagrama de Classes



# Especificação: Diagrama de Sequência – Inicializar interface Java





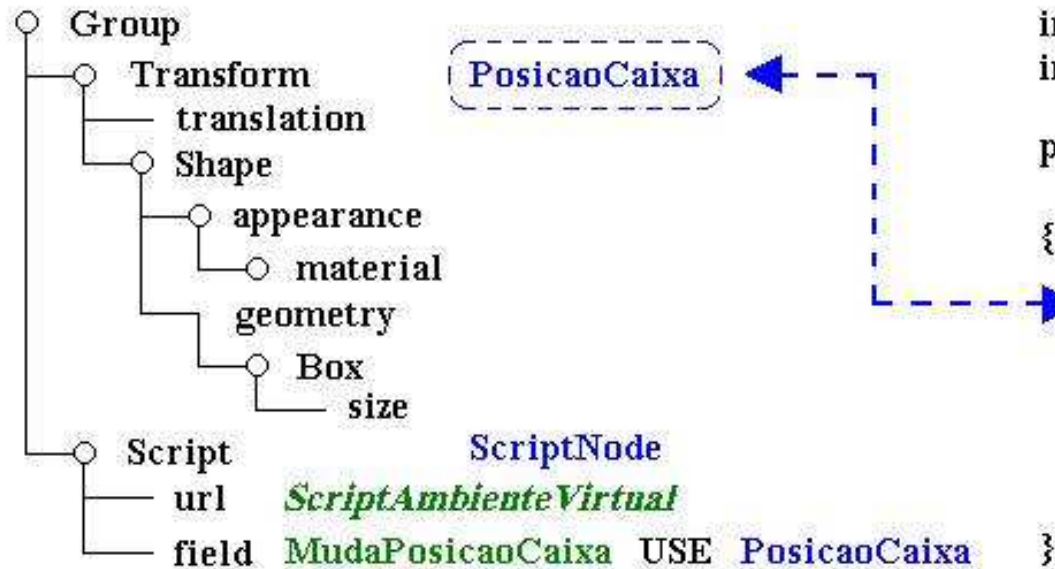
# Implementação:

## Técnicas e Ferramentas utilizadas

- Instalação do arquivo `jar` correspondente à tecnologia DIS-Java-VRML
- Netscape Communicator 4.77
- *Plugin* Cosmo Player que permite a visualização de ambientes VRML
- Plataforma de desenvolvimento Java JSDK 1.3

# Implementação: Comunicação Java x VRML

## Cena VRML



## Node DEFs

PosicaoCaixa

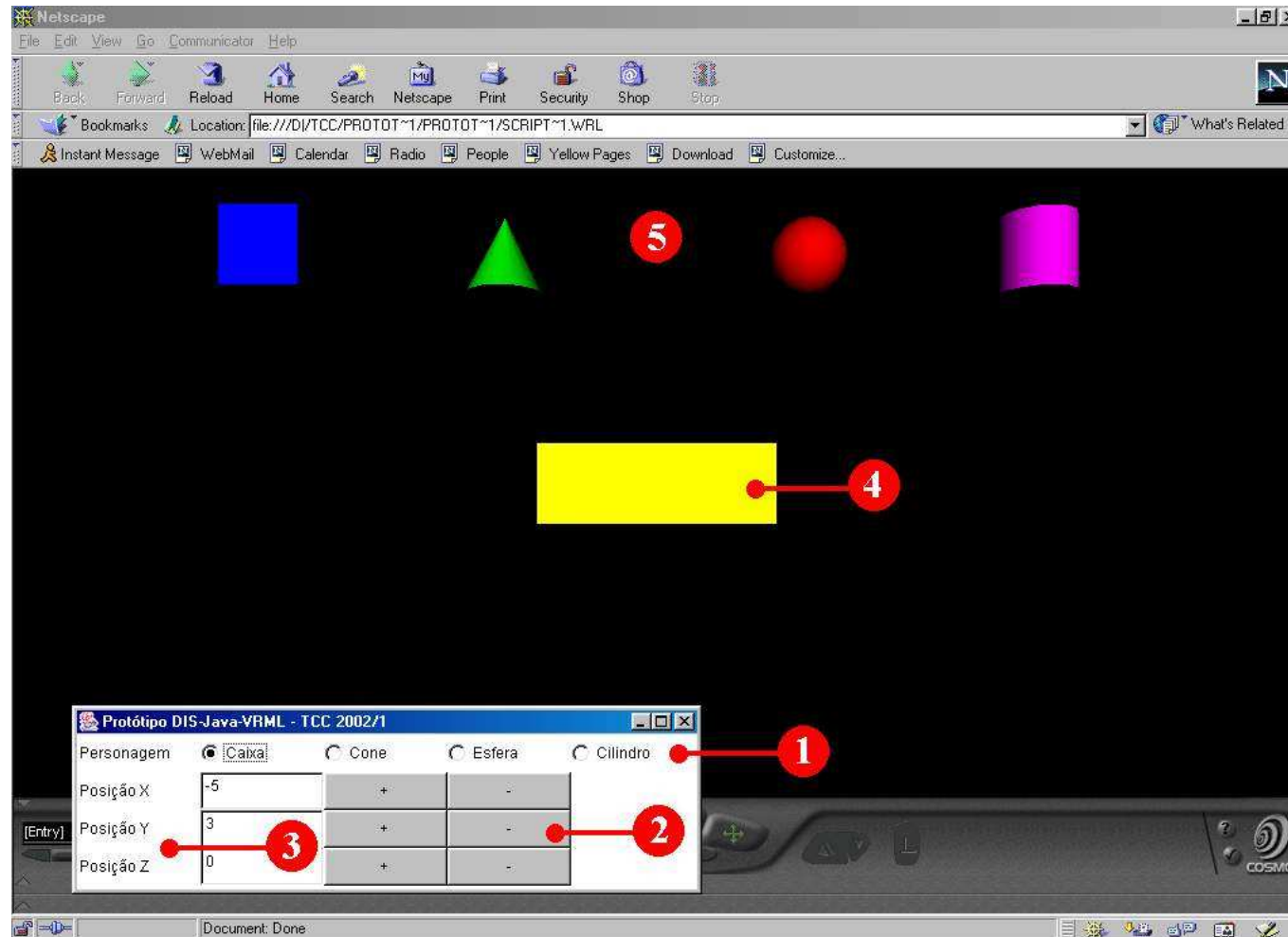
## Script Node Java

```
import vrml.*;
import vrml.field.*; import vrml.node.*;

public class
    ScriptAmbienteVirtual extends Script
    {
        field MudaPosicaoCaixa
        public void initialize ()
        { // método chamada ao abrir o
          // arquivo VRML
        }
    }
```

# Implementação:

## Operacionalidade da Implementação





# Conclusão

- Os objetivos foram alcançados
- São necessários cuidados especiais com a instalação dos softwares utilizados
- Pontos positivos e negativos





# Conclusão:

## Pontos positivos e negativos

Positivos	Negativos
Facilidade de uso do DIS através de suas classes já implementadas	Classes desenvolvidas especificamente para o Departamento de Defesa dos Estados Unidos
O envio de PDU's é realizado uma única vez através do endereçamento <i>broadcast</i>	O envio via <i>broadcast</i> pode ocasionar perda de PDU's e desempenho de rede
Classes do DIS foram desenvolvidas em Java, assim como o protótipo deste trabalho	
Existem classes em Java para trabalhar com a VRML	
É possível exibir o mundo virtual em um <i>browser</i> através de um <i>plugin</i> VRML	Interação do usuário com o mundo através do <i>plugin</i> causa problemas com o sistema operacional
Fácil entendimento do código VRML utilizando primitivas gráficas simples	
Grande variedade de exemplos disponíveis na instalação do DIS-Java-VRML	Dificuldades com a documentação de apoio do DIS-Java-VRML



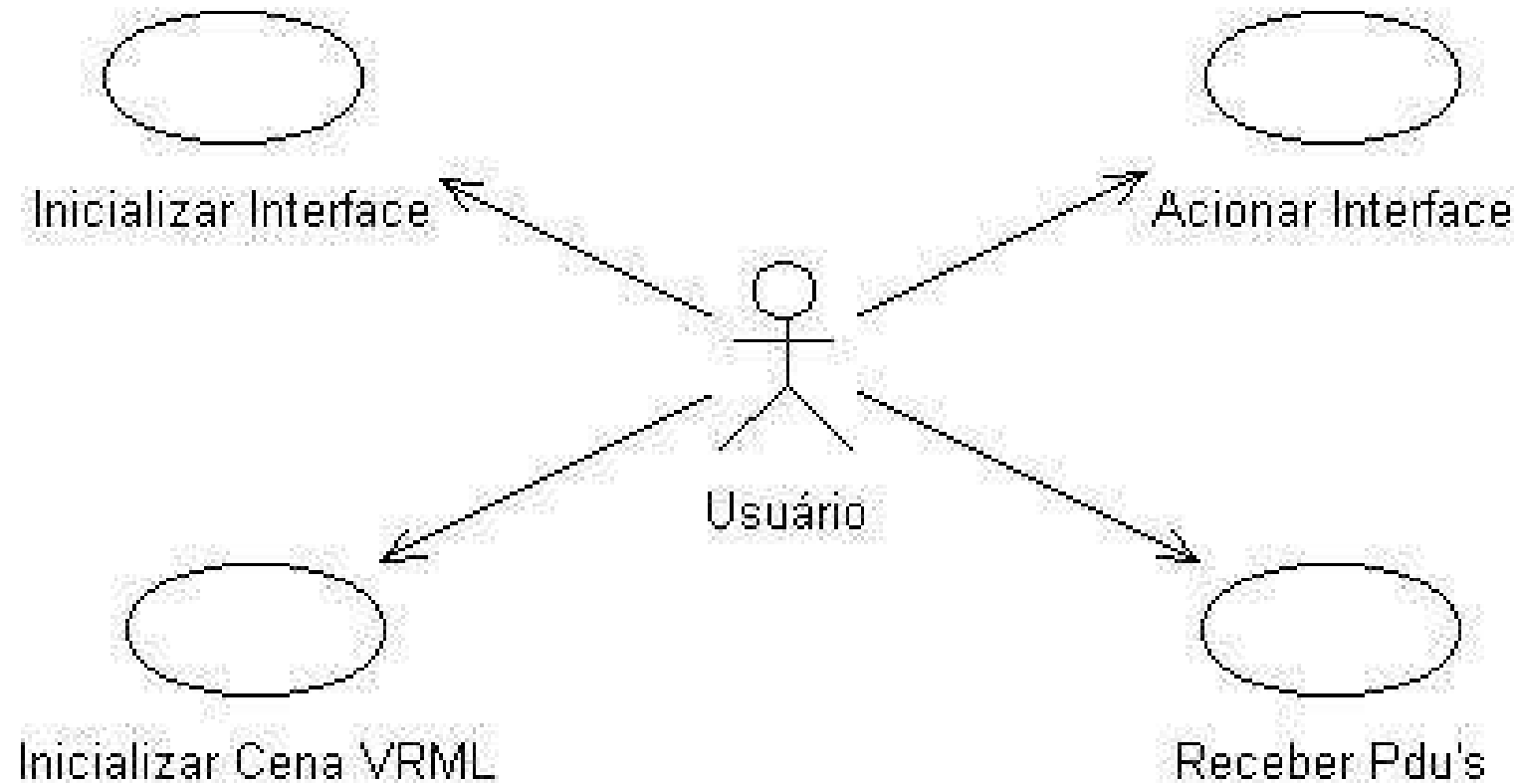
# Extensões

- Melhorar a aparência do ambiente virtual
- Permitir um número maior de participantes
- Melhorar o processo de comunicação
- Utilizar endereçamento multicast
- Comunicação entre o DIS e a tela em Java

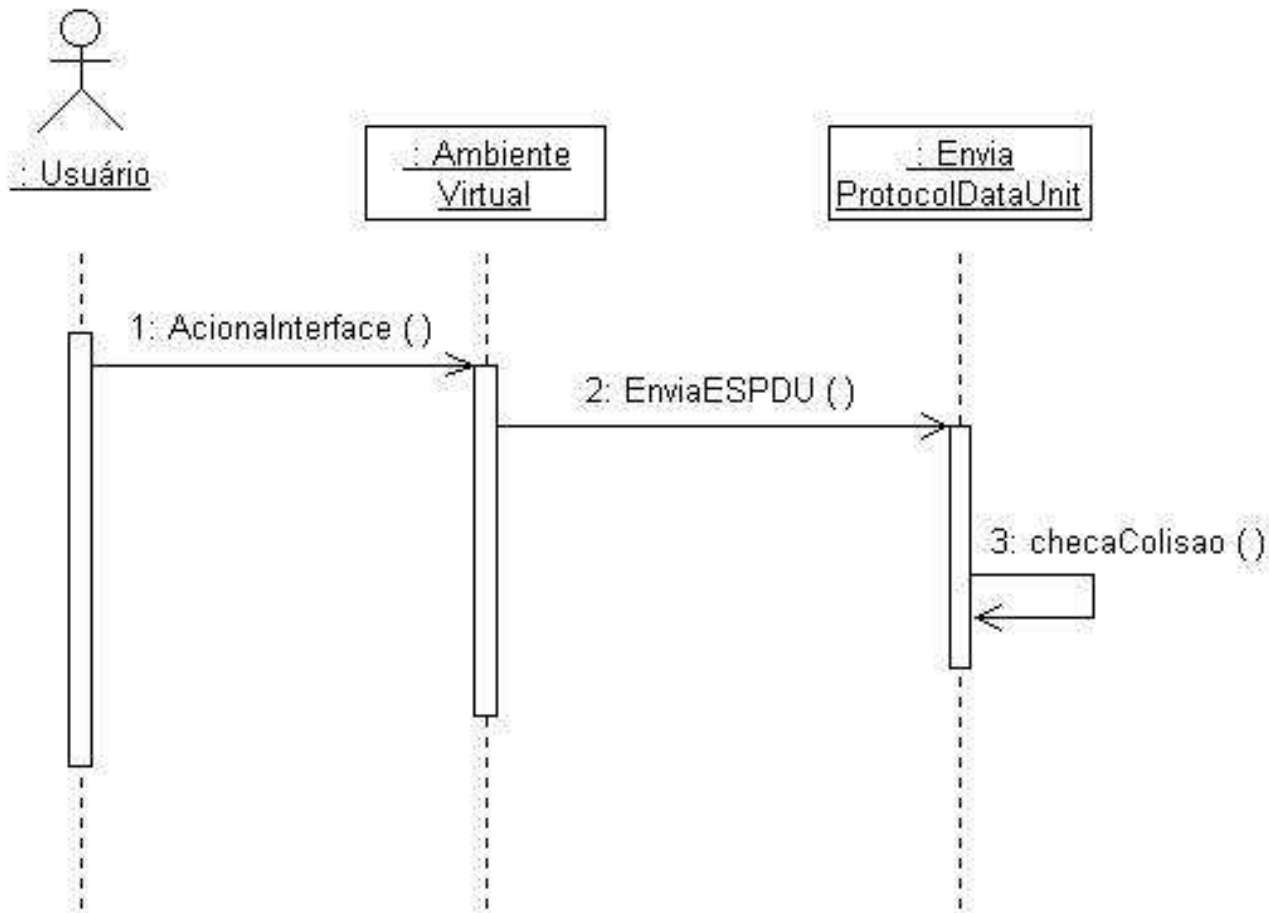


# Apresentação do Protótipo

# Especificação: *Use-Case*

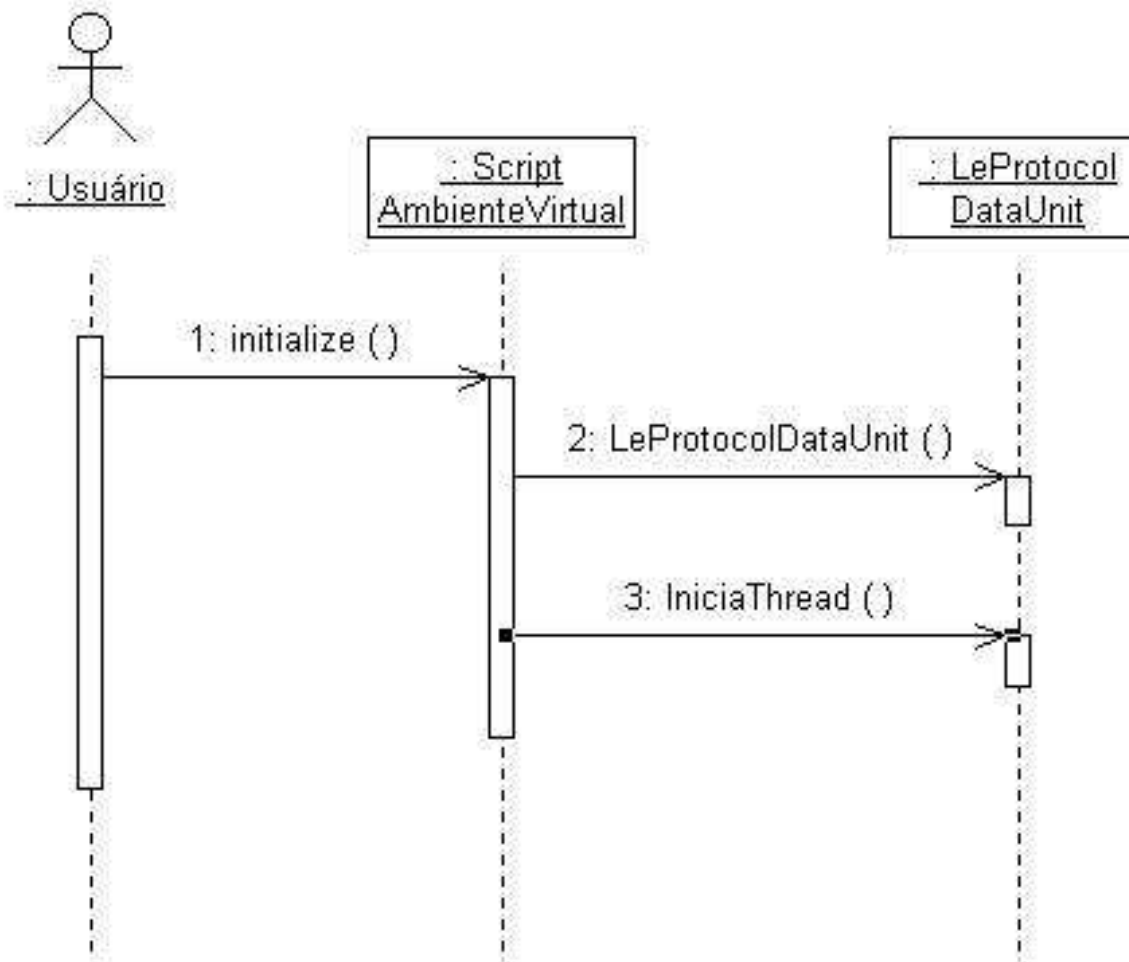


# Especificação: Diagrama de Sequência – Acionar interface Java



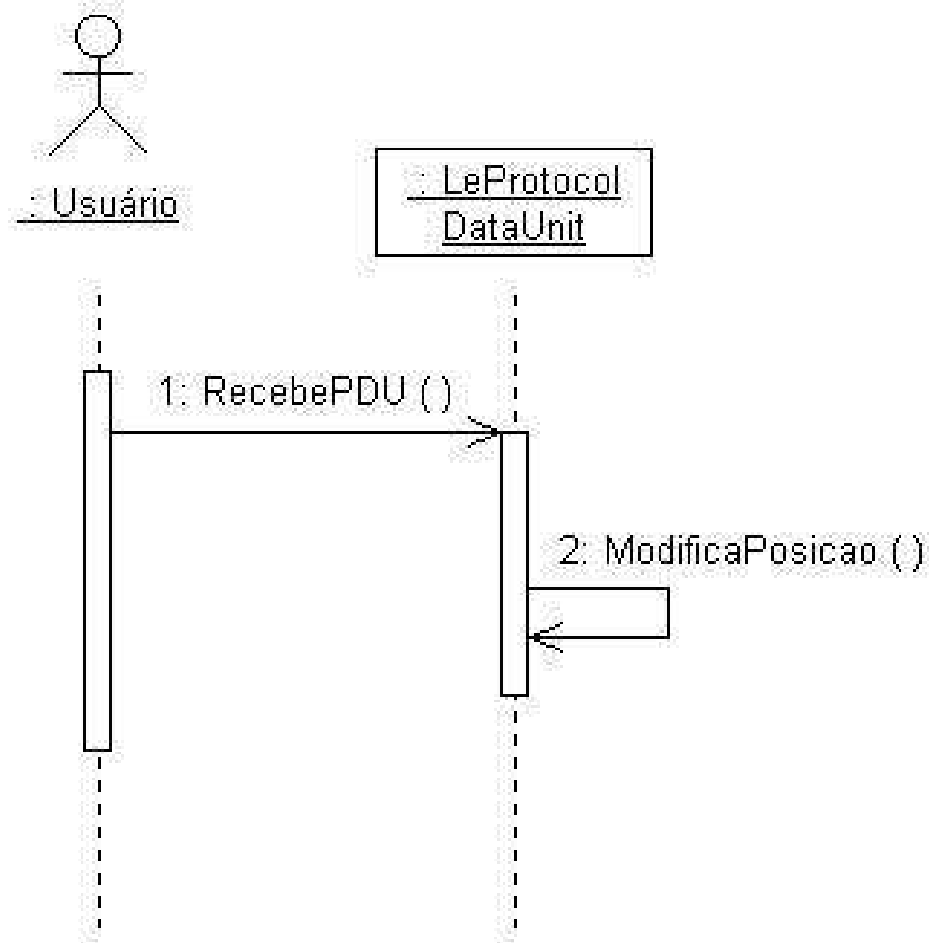
# Especificação:

## Diagrama de Sequência – Inicializar cena VRML



# Especificação:

## Diagrama de Sequência – Receber PDU's



# Extensão

