

# Curso Introdutório ao basAR

**Christopher Shneider Cerqueira**  
( [christophercerqueira@gmail.com](mailto:christophercerqueira@gmail.com)) e  
**Cláudio Kirner**  
([ckirner@gmail.com](mailto:ckirner@gmail.com)) .

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI  
2011

<https://sites.google.com/site/christophercerqueira/projetos/ear>

<http://www.realidadevirtual.com.br/>

# Guidelines

- Apresentação basAR (Exemplos e origem)
  - Entendendo o funcionamento
  - Alterações no exemplo.
  - Conclusões
- 
- Aplicação matemática, avaliação.

## Apresentando basAR

- RA
- Ferramentas de autoria
- Fonte de conteúdo
- basAR
- Exemplos de utilização
- Exemplo desta Apresentação.

# Realidade Aumentada

- Uma definição mais atualizada é: “realidade aumentada é uma **interface** baseada na **sobreposição de informações virtuais** geradas por computador (envolvendo imagens estáticas e dinâmicas, sons espaciais e sensações hápticas) **com o ambiente físico** do usuário, percebida **através de dispositivos** tecnológicos e usando as **interações naturais** do usuário, no mundo físico.” (KIRNER 2011)

# Ferramentas de autoria

- SACRA

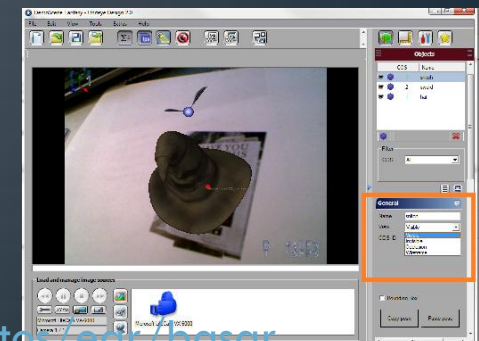
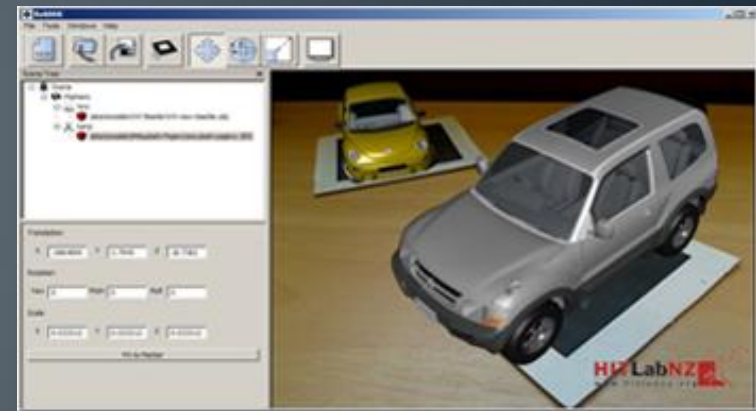
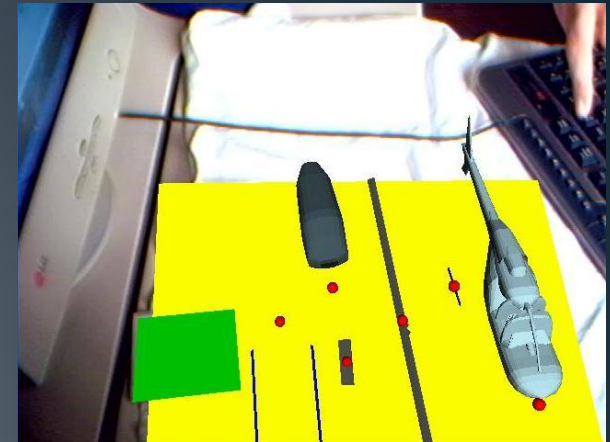
- <http://www.ckirner.com/sacra/>
- Colaboração
- Autoria
- Pontos de ação.

- BuildAR

- <http://www.buildar.co.nz/>
- Marker/ Markerless
- Vários formatos de arquivo

- metalO Unifeye Design

- <http://www.metaio.com/software/design/>
- Comportamento dos marcadores

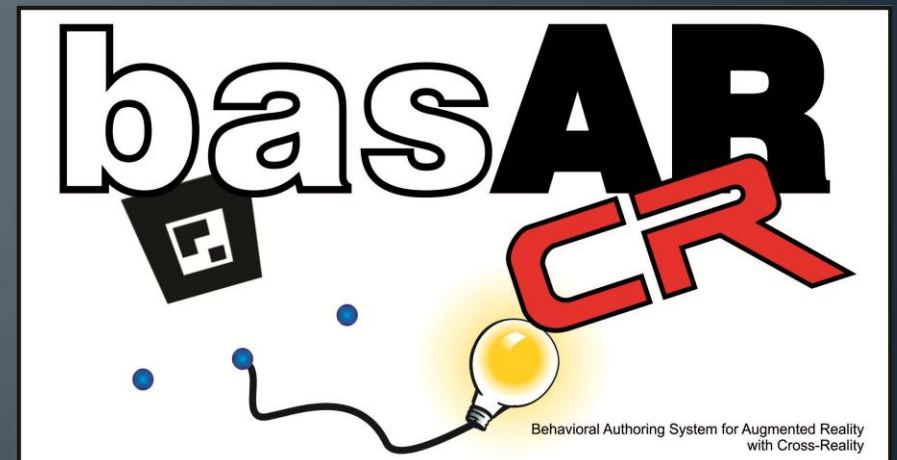
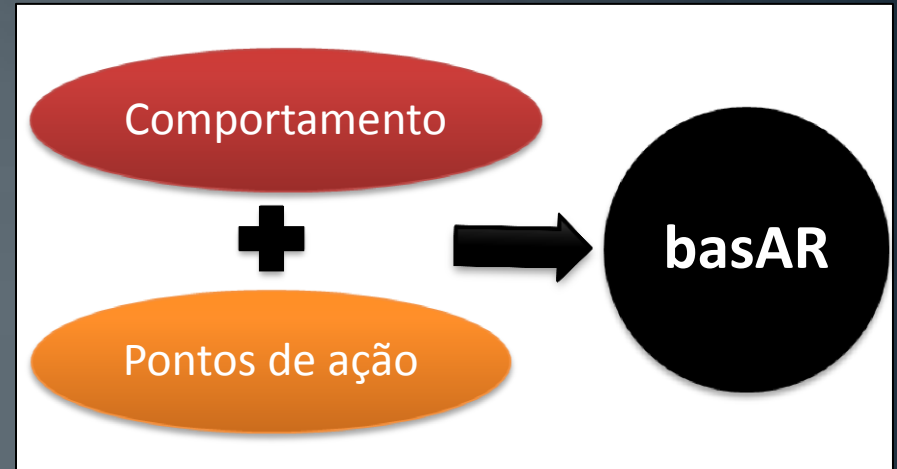


# Fontes de Conteúdo

- Repositório de imagens 3D
  - **Google SketchUp Warehouse** - <http://sketchup.google.com/3dwarehouse/>
- IDEs para desenvolvimento 3D
  - **3D Studio Max + Plugin Connection Extension**
  - Blender + Vivaty
  - Google SketchUp
- Repositório de sons
  - *JaySound* - <http://www.soundjay.com/>
  - *SoundBible* - <http://soundbible.com/>
  - *FindSounds* - <http://www.findsounds.com/>
- Programa de edição de áudio
  - Audacity - <http://audacity.sourceforge.net/?lang=pt>  
<https://sites.google.com/site/christophercerqueira/projetos/ear/basar>

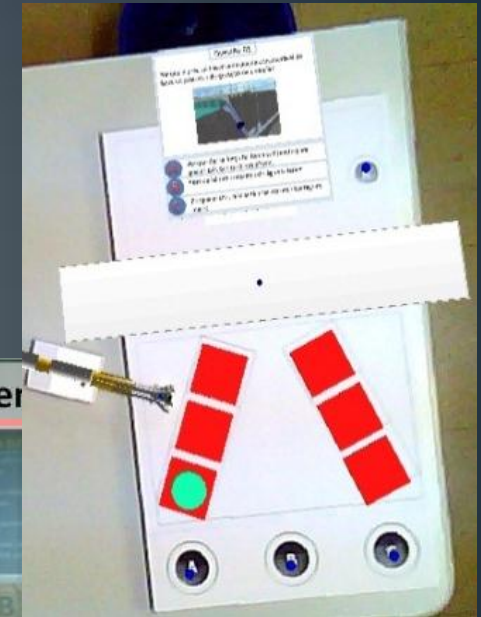
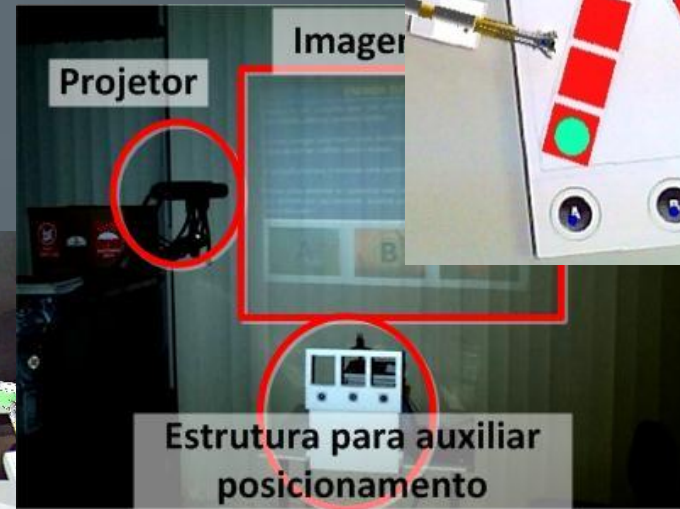
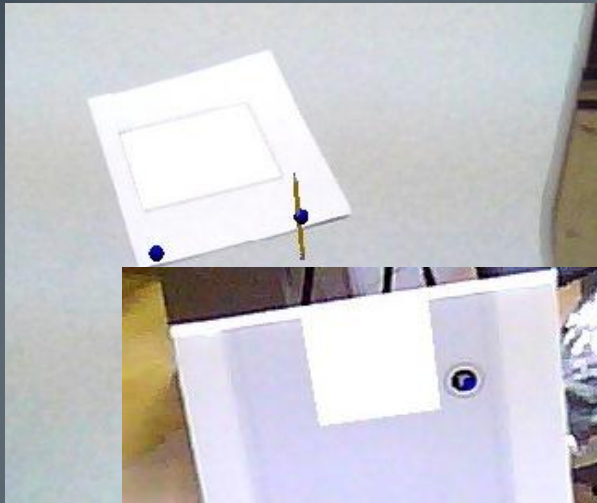
# basAR Behavioral Authoring System for Augmented Reality

- Sistema de Autoria Comportamental para Realidade Aumentada.
  - ARToolKit
  - Estrutura de comportamento
- COMPORTAMENTO DINÂMICO entre PONTOS DE AÇÃO





# Exemplos de utilização



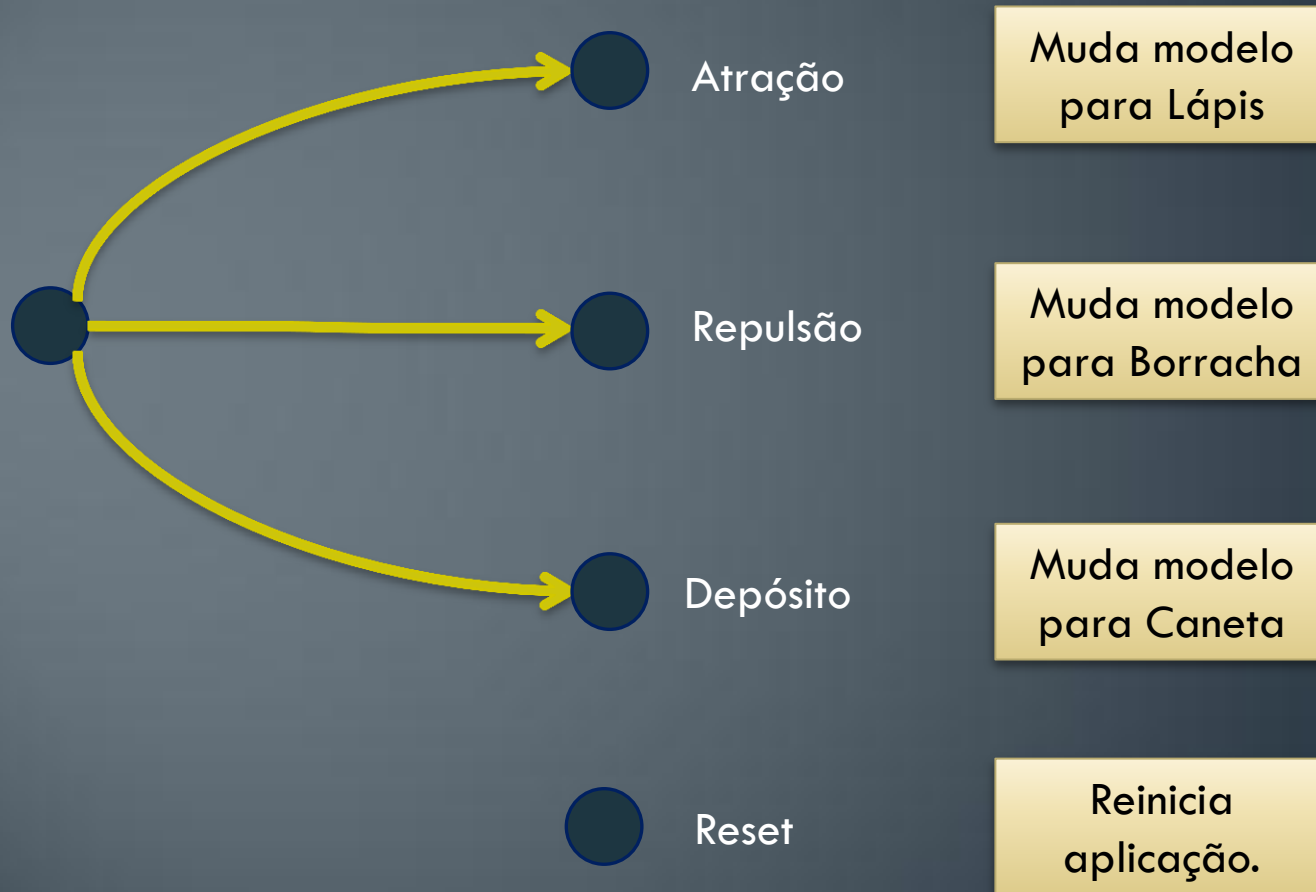


# Exemplo 2

- Fazer download do basAR e do pacote de modelos para o Exemplo 2 em um destes sites:
  - <https://sites.google.com/site/christophercerqueira/projetos/ear/basar>
- Pacote de modelos para o exemplo:
- Baixar o exemplo funcionando em:

<https://sites.google.com/site/christophercerqueira/projetos/ear/basar>

# Exemplo básico



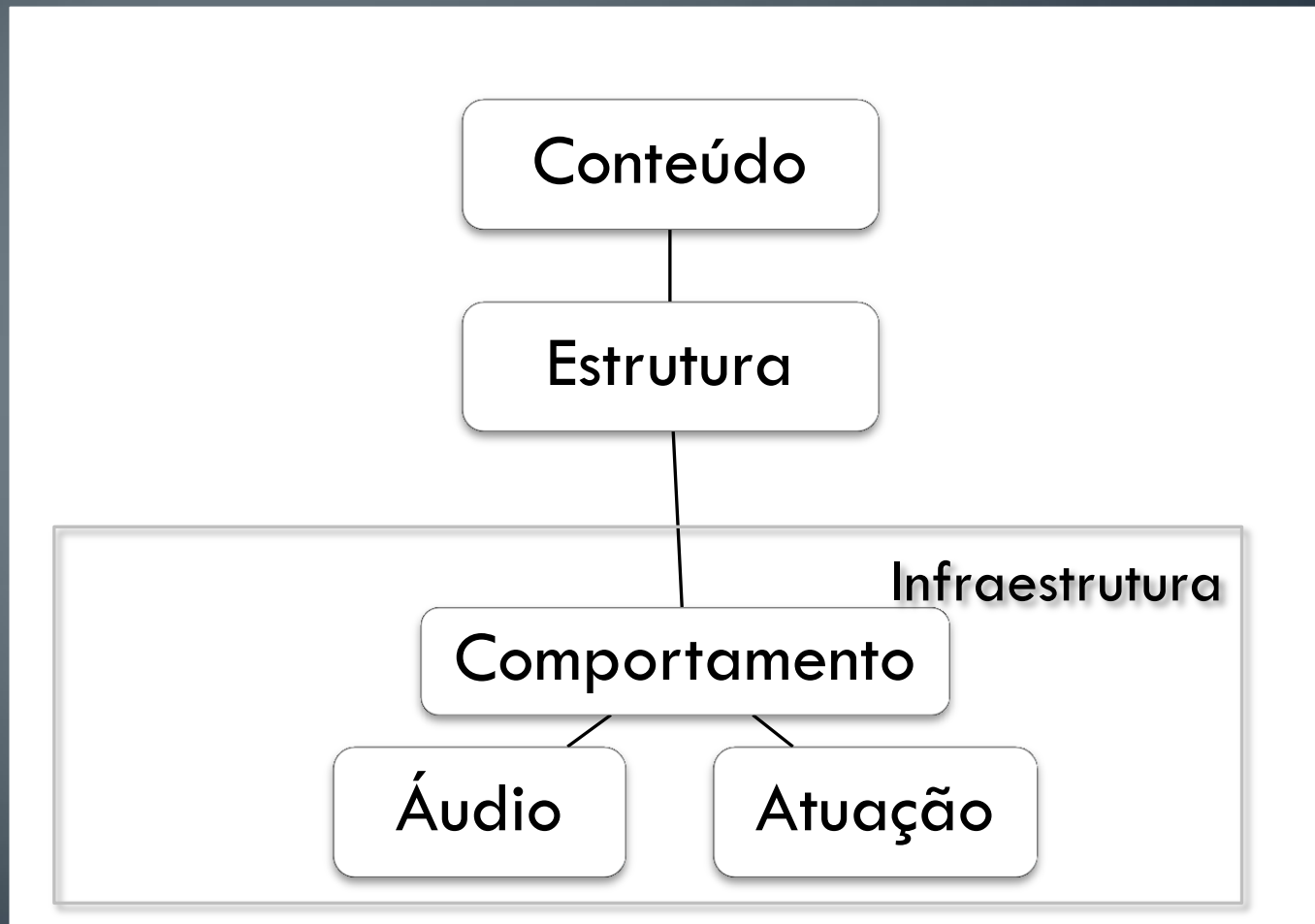
## Entendendo o funcionamento

- Infraestrutura
- Atuação
- Estrutura
- Conteúdo
- Comportamento

# Definições iniciais

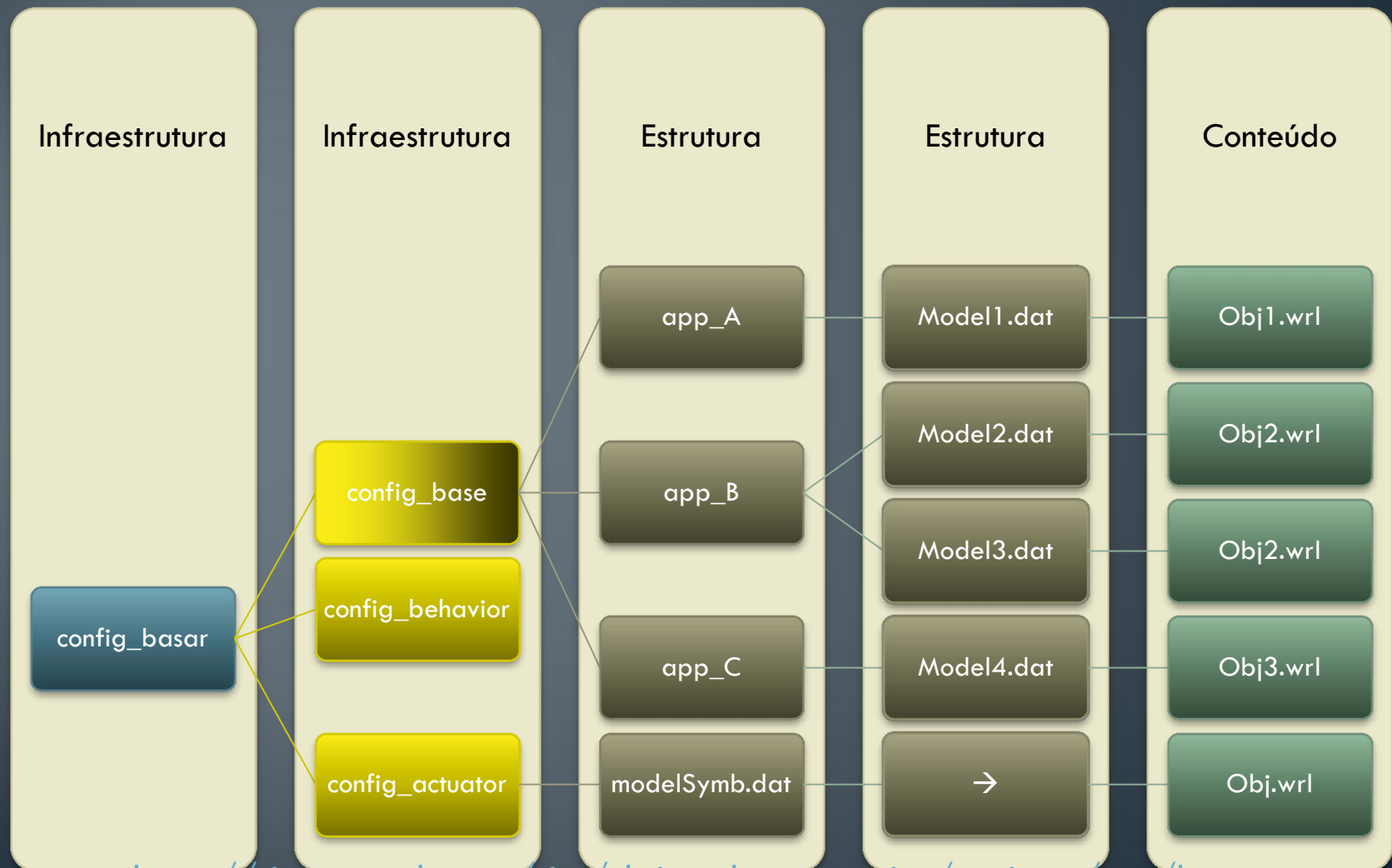
- **Infraestrutura:** como é realizada a correlação entre o mundo real e o mundo virtual.
- **Atuação:** como é realizada a interação do usuário com o sistema.
- **Estrutura:** localização, orientação e escala dos objetos virtuais.
- **Conteúdo:** quais objetos são utilizados.
- **Comportamento:** forma como a estrutura se comporta com as ações de atuação do usuário.

# Organização



- Como é construída a aplicação???????
- Arquivo de descrição editáveis.

# Arquivos de descrição



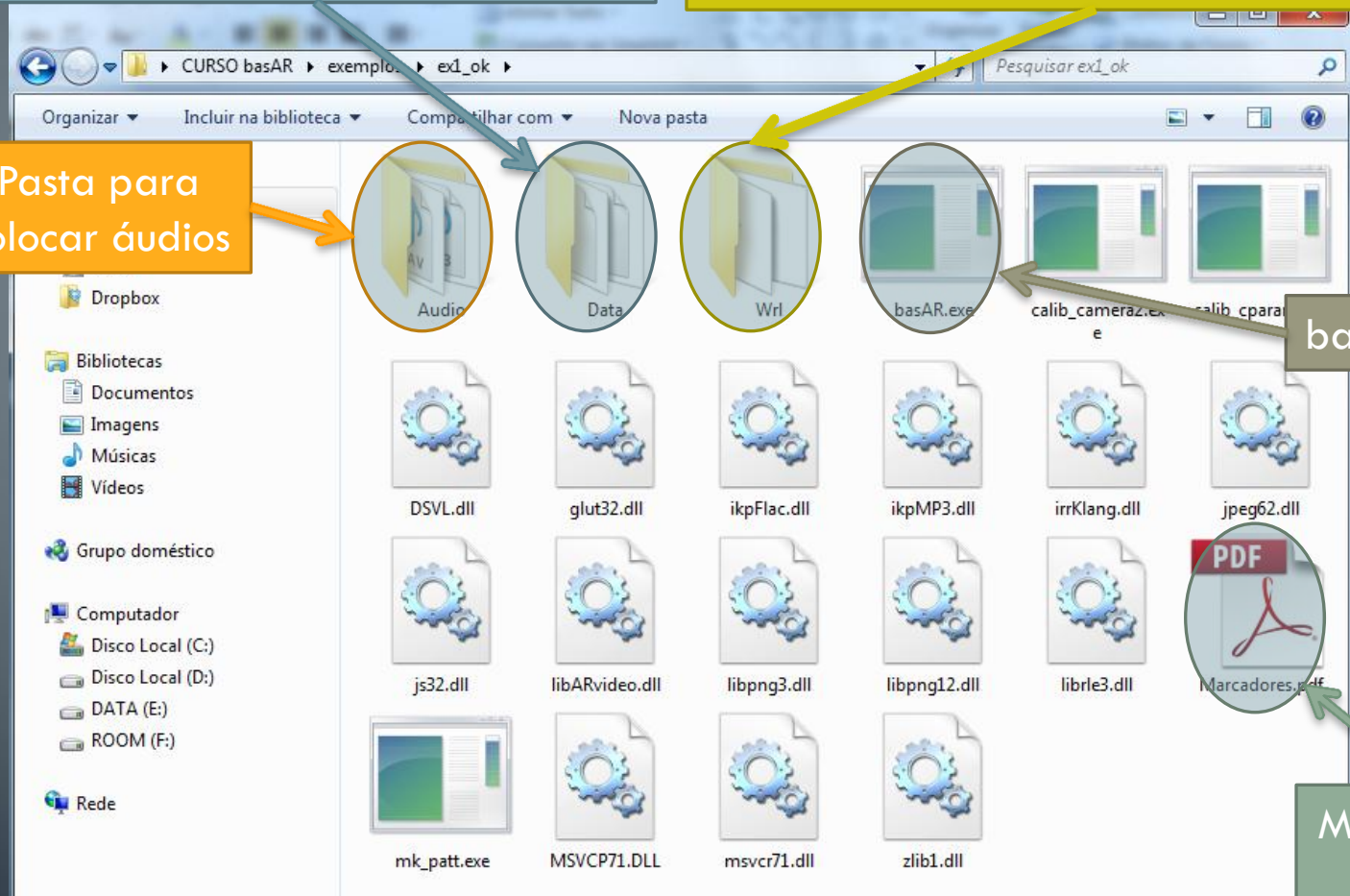


# Diretórios e arquivos do basAR

Pasta com arquivos de infraestrutura e estrutura

Pasta com modelos e arquivos de estrutura que descrevem o modelo

Pasta para colocar áudios



basAR

Marcadores  
padrão.

# Infraestrutura

# Configurando Infraestrutura

- Configurar a infraestrutura envolve 3 passos:
  - Definir itens genéricos e o relacionamento com o mundo real.
  - Configurar a base da área de trabalho.
  - Configurar artefato de interação.

# Configurando Infraestrutura – Passo 1

- O primeiro passo para configurar uma aplicação é definir os componentes básicos que fazem parte.
- No basAR, são necessários inicializar componentes genéricos e quais serão as bases e atuadores.
- Neste exemplo são utilizados **uma base** que relaciona o mundo real ao mundo virtual e **um artefato de atuação**.
- O primeiro arquivo de descrição a ser configurado é o data/**config\_basar**.

# Exemplo: config\_basar

## Exemplo do basAR

```
VRML wrl/action/ballBlue.dat
VRML Wrl/Action/ballGreen.dat
VRML Wrl/Action/ballRED.dat
VRML Wrl/action/tampa.dat
Audio/explosion.wav 0.5
```

```
Audio/backTrack.mp3 LOOP 0.3
Audio/bell.wav        ONCE 0.5
```

Data/config\_behavior

1

Data/config\_base

1

ARTKSM Data/config\_transport

Field
AppName
StdAwayModel
StdCorrectModel
StdWrongModel
StdMarkerCover
StdErrorSound
BackTrack
StartSound
StateMachine
NumBases
Bases
NumActuators
Actuator

# Configurando Infraestrutura – Passo 2

- O segundo passo de configurar infraestrutura é definir como é a fonte de dados de infraestrutura.
- No momento só é possível marcador simples. (1 marcador)
- Além de definir algumas características padrões desta fonte de dados (base ou referência).
- O arquivo que configura a base é o arquivo data/**config\_base** como definido no arquivo data/config\_basar.
- O arquivo config\_base define infraestrutura e estrutura. Vamos configurar a parte de estrutura dele em outro momento.

## Configurando Infraestrutura – Parte 2





# Foto do exemplo

# Exemplo: config\_base

**BASE1**

**ARTKSM**

**Data/Markers/base.patt**

**80.0**

**0.0 0.0**

**USE\_DEFAULT**

**NO\_VISIBLESOUND**

**USE\_DEFAULT**

**wrl/action/status.dat**

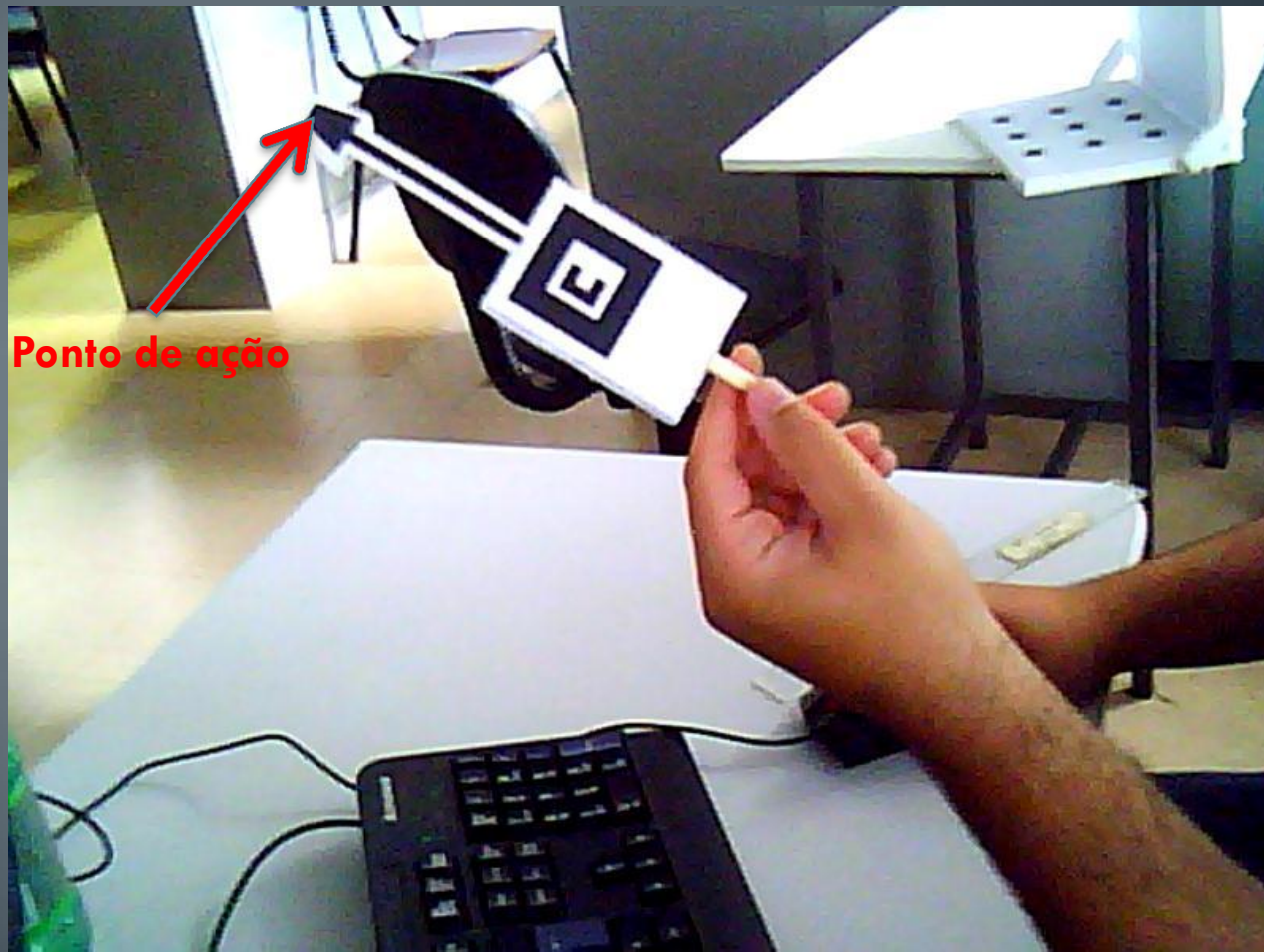
**0**

Field
BaseName
If using ARToolKit Single Marker Source
SourceType
PatternSource
PatternWidth
PatternCenter
PatternCover
VisibleSound
ErrorSound
BaseStatus
NumPoints

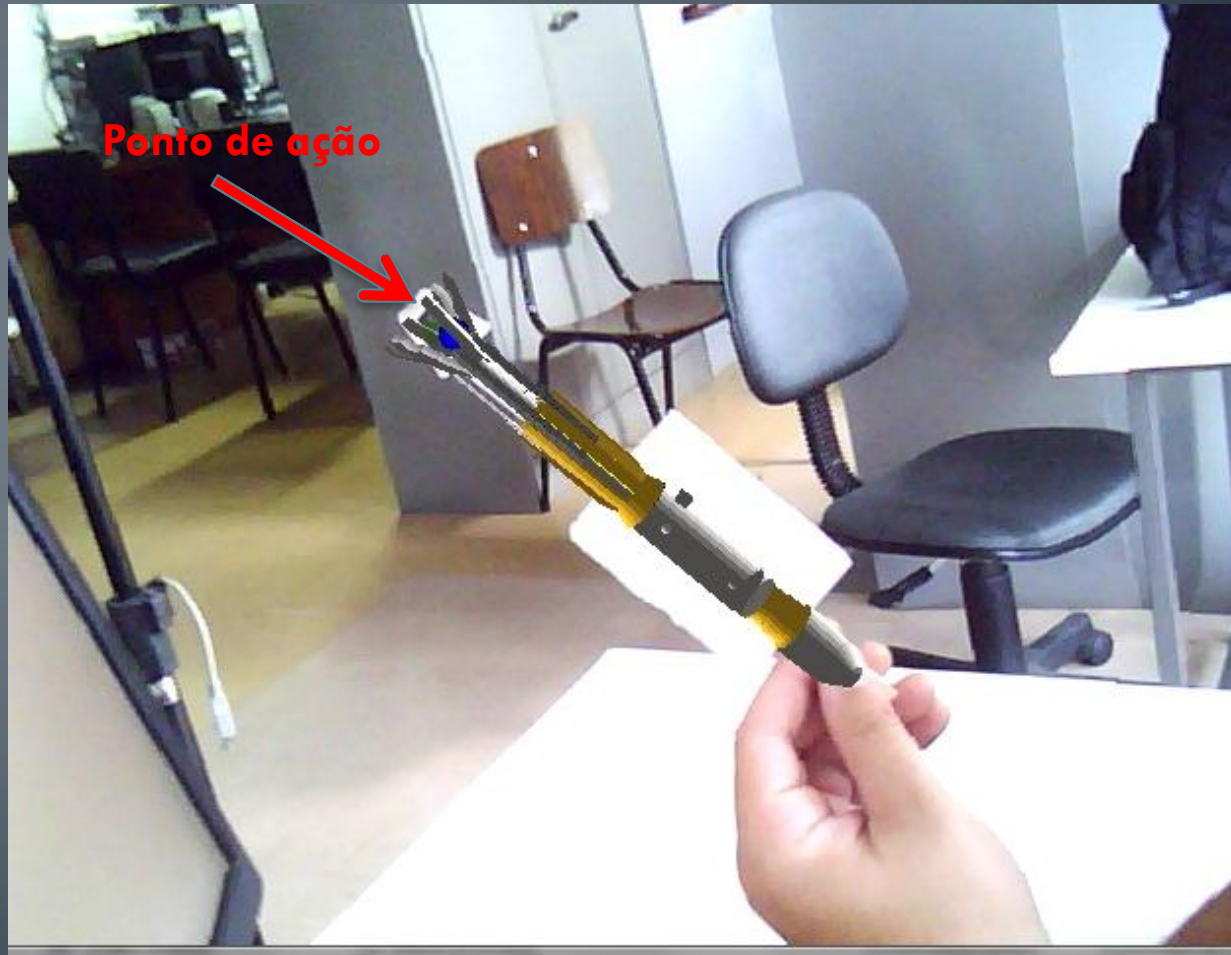
# Configurando Infraestrutura – Passo 3

- O terceiro passo na configuração da infraestrutura é a definição do artefato de atuação.
- No momento só é possível marcador simples. (1 marcador)
- O artefato tem um modelo simbólico que define como o atuador é mostrado para a abstração da aplicação.
- O arquivo que configura a base é o arquivo data/**config\_transport** como definido no arquivo data/config\_basar.

# Configurando Infraestrutura – Parte 3



# Configurando Infraestrutura – Parte 3



# Exemplo: config\_actuator

ARTKSM1

Data/Markers/pa.patt

37.0

0.0 0.0

VRML Wrl/action/tampaAct.dat

VRML wrl/Action/pa.dat

DEFAULT\_IPOINT

20.0 0.0 0.0

400.0

Field
ActuatorName
If using ARToolKit Single Marker Source
PatternSource
PatternWidth
PatternCenter
PatternCover
SymbolicModel
PointModel
PointPosition
PointRadius

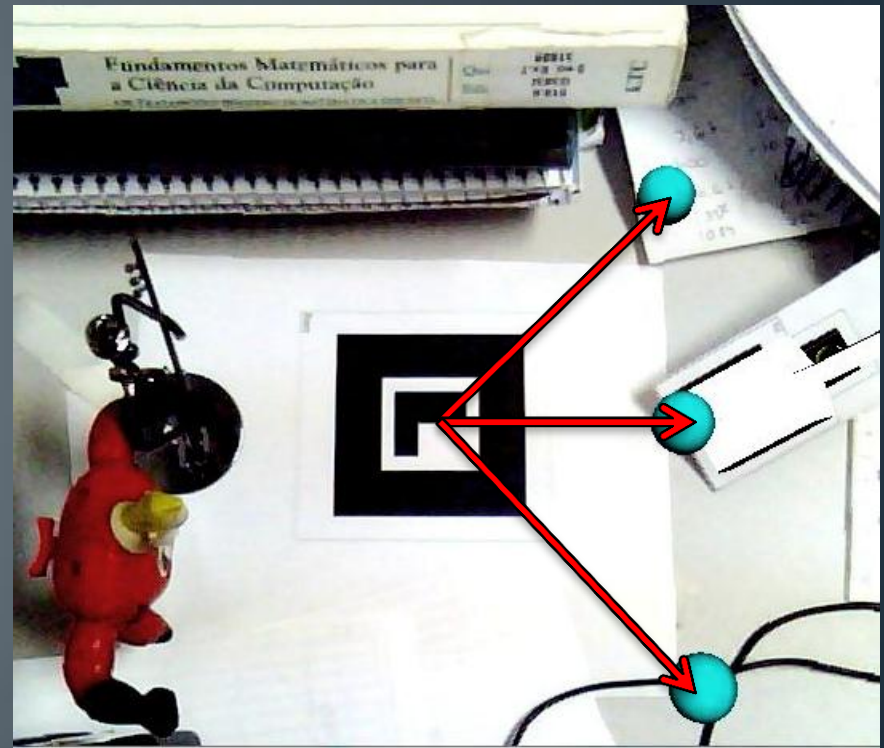
# Configurando estrutura

- Configurar a estrutura envolve o posicionamento dos objetos virtuais em relação a fonte de infraestrutura.
- No caso a fonte de dados da infraestrutura é o marcador de base (ou referência)
- Configurar a estrutura envolve configurar pontos de ação.
- Entender:
  - Zona reativa.
  - Inserir modelos
  - Pivot de movimentação.



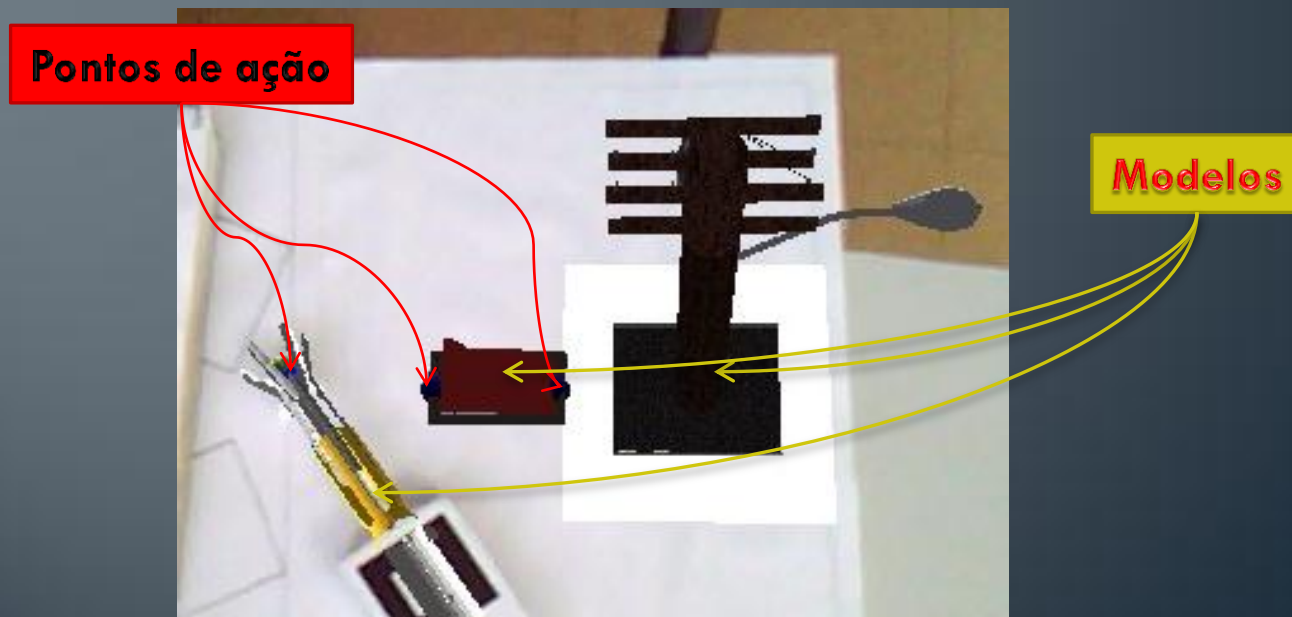
# Pontos de ação

- **Zonas reativas** em relação a uma referência que possuem alguma característica.
- Pontos possuem **esferas de ação** e **modelos associados**.
- A **interação** entre os pontos de ação definem os comportamentos que podem ser tomados por uma aplicação.



# Componentes do ponto de ação.

- Ponto de ação (BALL)
  - Representação do ponto de posicionamento da ação. Não condiz com o tamanho do raio de ação do comportamento.
- Modelos associados (OBJECTS)
  - Modelos 3D associados ao ponto de ação.



# Pivot

- Dois braços de movimentação.
  - Centro do marcador → Centro do ponto de ação
  - Centro do ponto de ação → Centro do modelo 3D.

# Estrutura

# Configurando estrutura

- A estrutura é configurada em 3 passos.
  - Criação dos pontos na base
  - Descrição da lista de modelos dos pontos
  - Descrição do arquivo a ser carregado e ajuste de pivot.

# Configurando estrutura – Parte 1

- O primeiro passo da configuração da estrutura é criar na base os pontos que serão utilizados.
- Modificando o arquivo data/**config\_base**.
- Inserir os pontos e dizer a posição, qual arquivo das esferas de ação e a lista de modelos associados.
- Neste exemplo serão usados **5 pontos**.
  - 1 ponto para movimentação, que tem 3 modelos (borracha, caneta e lápis)
  - 1 ponto de atração, que tem 1 modelo (placa escrito Atração).
  - 1 ponto de repulsão, que tem 1 modelo (placa escrito Repulsão).
  - 1 ponto de depósito, que tem 1 modelo (placa escrito Depósito).
  - 1 ponto de reset, que tem 1 modelo (placa escrito Reset).

# Exemplo: config\_base (Parte 2 – Estrutura)

...

5

Objeto

DEFAULT\_IPOINT

Data/app\_objeto

200.0 0.0 0.0

0.0 0.0 0.0

1 1 1

500.0

Atracao

DEFAULT\_IPOINT

Data/app\_at

100.0 -100.0 0.0

0.0 0.0 0.0

1 1 1

500.0

Repulsao

DEFAULT\_IPOINT

Data/app\_rp

100.0 0.0 0.0

0.0 0.0 0.0

1 1 1

500.0

Deposito

DEFAULT\_IPOINT

Data/app\_dp

100.0 100.0 0.0

0.0 0.0 0.0

1 1 1

500.0

Reset

DEFAULT\_IPOINT

Data/app\_rs

100.0 200.0 0.0

1 1 1

500.0

**Action Point  
structure**

**PointName**

**ActionModel**

**ObjectModels**

**StartTranslation**

**StartRotation**

**StartScale**

**StartRadius**



# Configurando estrutura – Parte 2

- O segundo passo é povoar os arquivos com a lista de modelos dos pontos. (Caso o ponto tenha objetos – **NO\_OBJECT** (não tem modelos 3D)).
- Os nomes dos arquivos estão descritos no arquivo data/config\_base:
  - **Data/app\_objeto**
  - **Data/app\_at**
  - **Data/app\_rp**
  - **Data/app\_dp**
  - **Data/app\_rs**
- No momento basAR suporta modelos **VRML**, estáticos ou animados, com textura **.gif** .

# Exemplo: app\_XXX (Listas de objetos)

```
#Data/app_object
```

```
3
```

```
MODEL3D VRML Wrl/caneta.dat
```

```
MODEL3D VRML Wrl/borracha.dat
```

```
MODEL3D VRML Wrl/lapis.dat
```

Field
NumObjects
Object structure
Object

```
#Data/app_at
```

```
1
```

```
MODEL3D VRML Wrl/placaAt.dat
```

```
#Data/app_dp
```

```
1
```

```
MODEL3D VRML Wrl/placaDp.dat
```

```
#Data/app_rp
```

```
1
```

```
MODEL3D VRML Wrl/placaRp.dat
```

```
#Data/app_rs
```

```
1
```

```
MODEL3D VRML Wrl/placaRs.dat
```

# Configurando estrutura – Parte 3

- O terceiro passo da configuração da estrutura é a indicação do nome do arquivo VRML que vai ser carregado e a relação entre o ponto de ação e o modelo (pivot)
- No basAR os arquivos que configuram os modelos (.dat) seguem a filosofia do ARToolKit e do SACRA e localizam-se na pasta WRL.
- Cada modelo tem que ser descrito por um arquivo (.dat)

# Exemplo: xxx.dat

#caneta.dat

pen.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

#placaAt.dat

at.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

#lapis.dat

pencil.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

#placaRp.dat

rp.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

#borracha.dat

eraser.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

#placaDp.dat

dp.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

#placaRs.dat

rs.wrl

0.0 0.0 00.0

0.0 0.0 0.0

0.25 0.25 0.25

Field

WRLFileAddress

Translation

Rotation

Scale

# Fim da configuração de estrutura

## Conteúdo

# Configurando conteúdo

<https://sites.google.com/site/christophercerqueira/projetos/ear/basar>

# Até agora tudo bem?



# Estrutura de itens atual



## Configurando comportamento

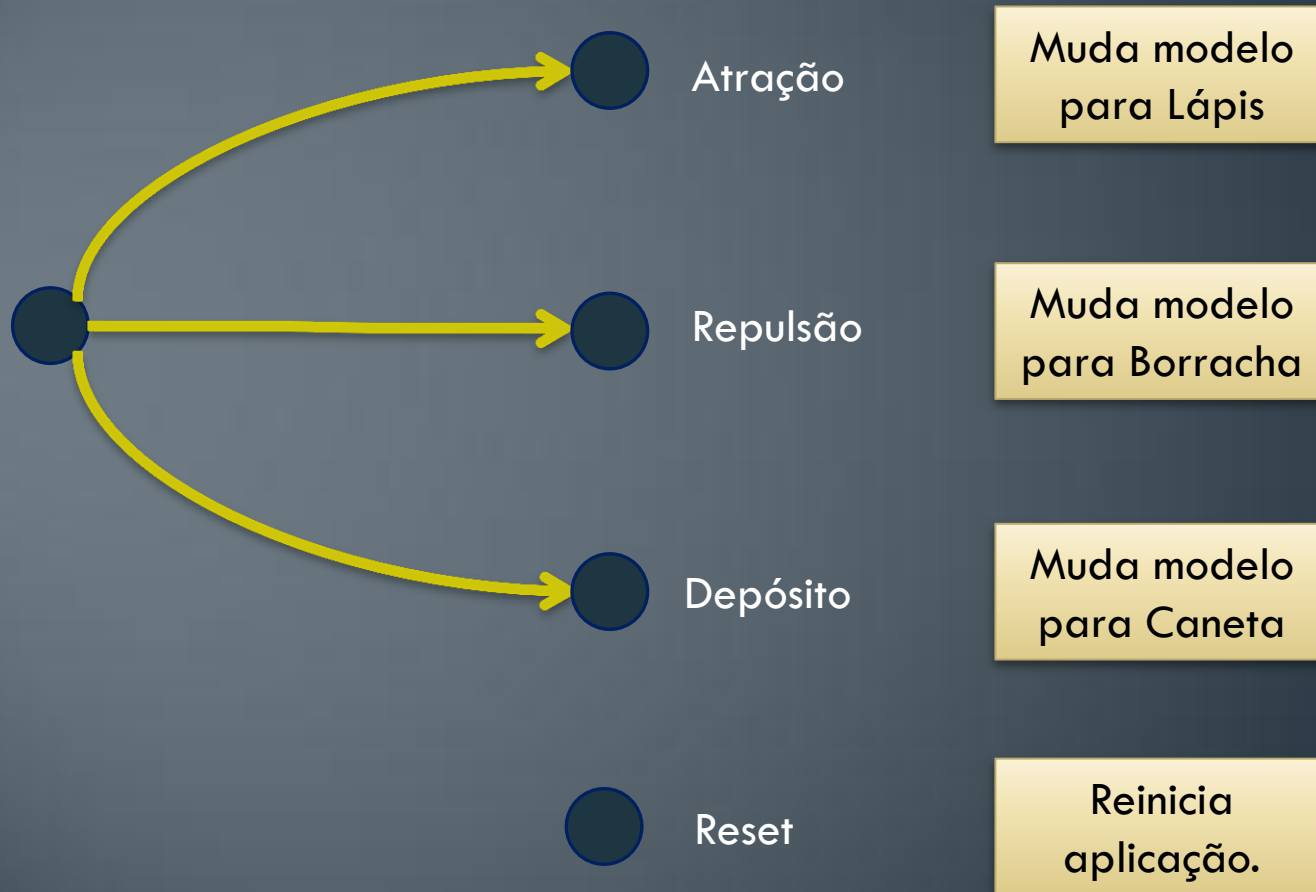
# Configurando comportamento

- O diferencial do **basAR** para as outras ferramentas é a configuração de **COMPORTAMENTO DINÂMICO** entre **PONTOS DE AÇÃO**.
- Os comportamentos são configurados para cada ponto independentemente.
- **Comportamento = Modo de exibição + Ação**

# Configurando comportamento

- **Deve-se conhecer o comportamento da aplicação.**
- O comportamento deste exemplo:
  - 1 Ponto pode ser movido livremente, e inicia com o modelo de uma caneta. Se o usuário, movimentando o ponto com o atuador, colidir com os pontos:
    - De atração: o ponto em movimento recebe as características do ponto de colisão. E muda o modelo do ponto em movimento para um lápis
    - De repulsão: o ponto em movimento vai para o local que iniciou o movimento. E muda o modelo do ponto em movimento para uma borracha.
    - De depósito: o ponto é depositado no local se o usuário “tampar” o marcador de atuação. Muda o modelo do ponto em movimento para uma caneta.
  - “Tocando” o ponto de reset com o atuador, faz com que a aplicação seja reiniciada.

# Configurando comportamento

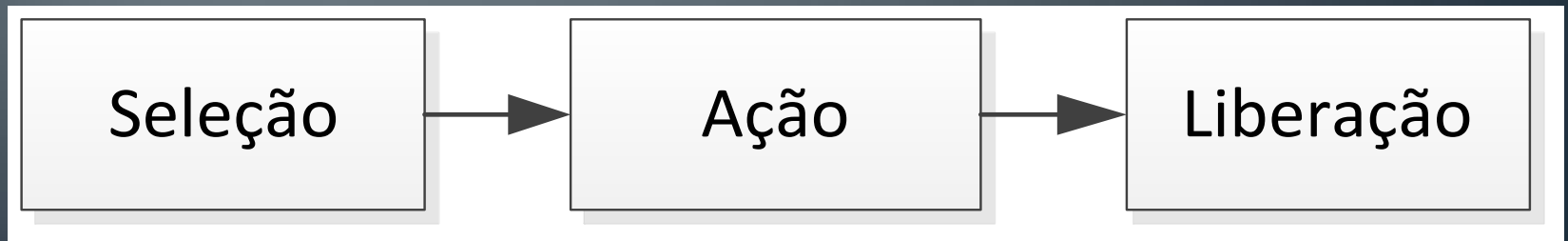


# Configurando comportamento

- **Necessário entender:**
  - O que é comportamento.
  - Mapa de comportamento.
  - Estados do cenário.
  - Máquina de estados.
  - Transformação de comportamento e máquina de estados em uma estrutura de comandos.
    - Modos de exibição
    - Tipos de comandos
    - Tipos de estados

# O que é Comportamento?

- Comportamento é a **resposta** do sistema **ao estímulo** do usuário em um **ponto de ação**.
- Um ponto deve ser selecionado para que uma ação ocorra, e ao fim desta ação o ponto deve ser liberado



# Comportamento

- Mapa de comportamento
- Movimentação ou configuração
- Zona reativa interfere na forma como o ponto é liberado na movimentação.

Selection

Collision between actuator and action point

If selectable

Point selected

Manipulation

Move action point

Configure System

System Controlled

User controlled

Release

Attract point

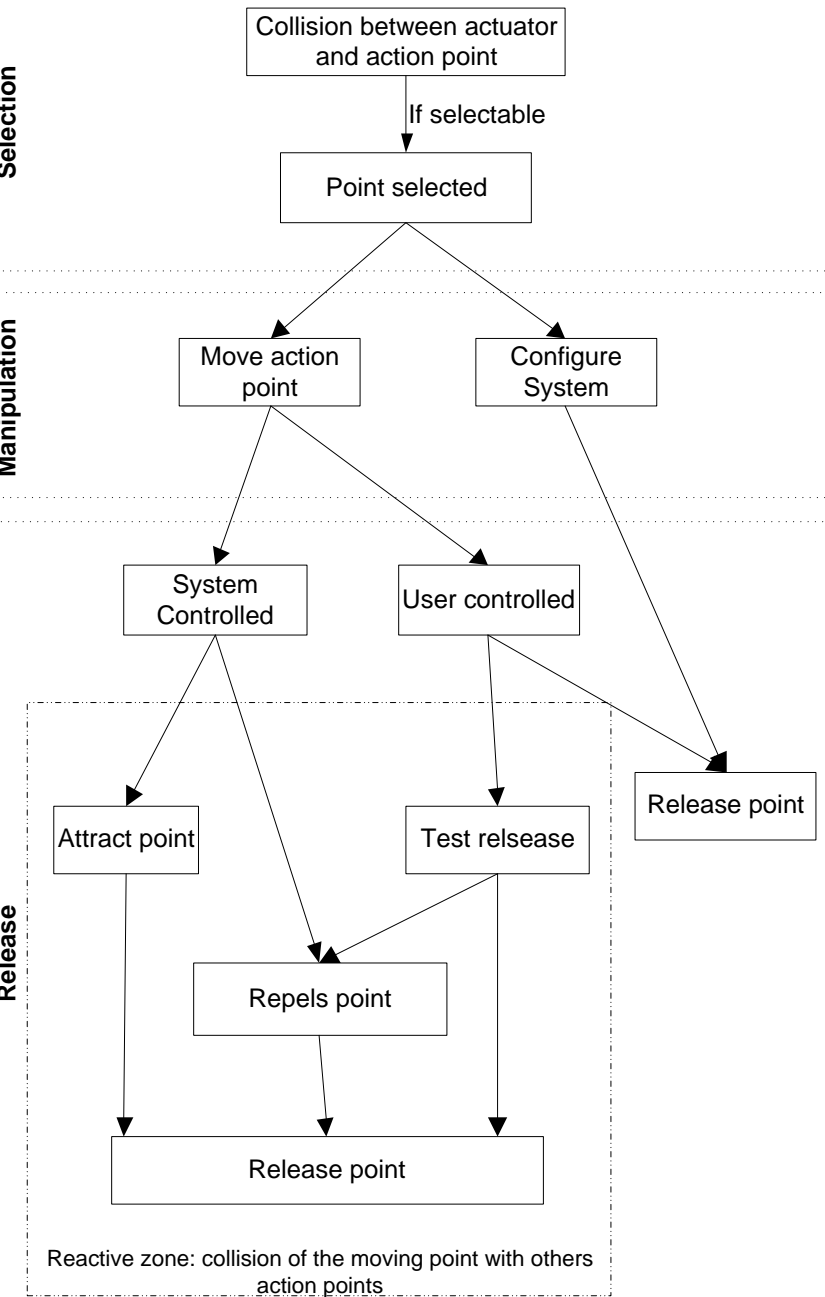
Test release

Release point

Repels point

Release point

Reactive zone: collision of the moving point with others action points





# Estados

- Configuração do ambiente em determinado momento.
  - Conjunto de características e comportamentos dos pontos de ação.

## Estado 1

Exibe caneta e pode mover.

Exibe placa Atrair e atrai.

Exibe placa Repelir e repele.

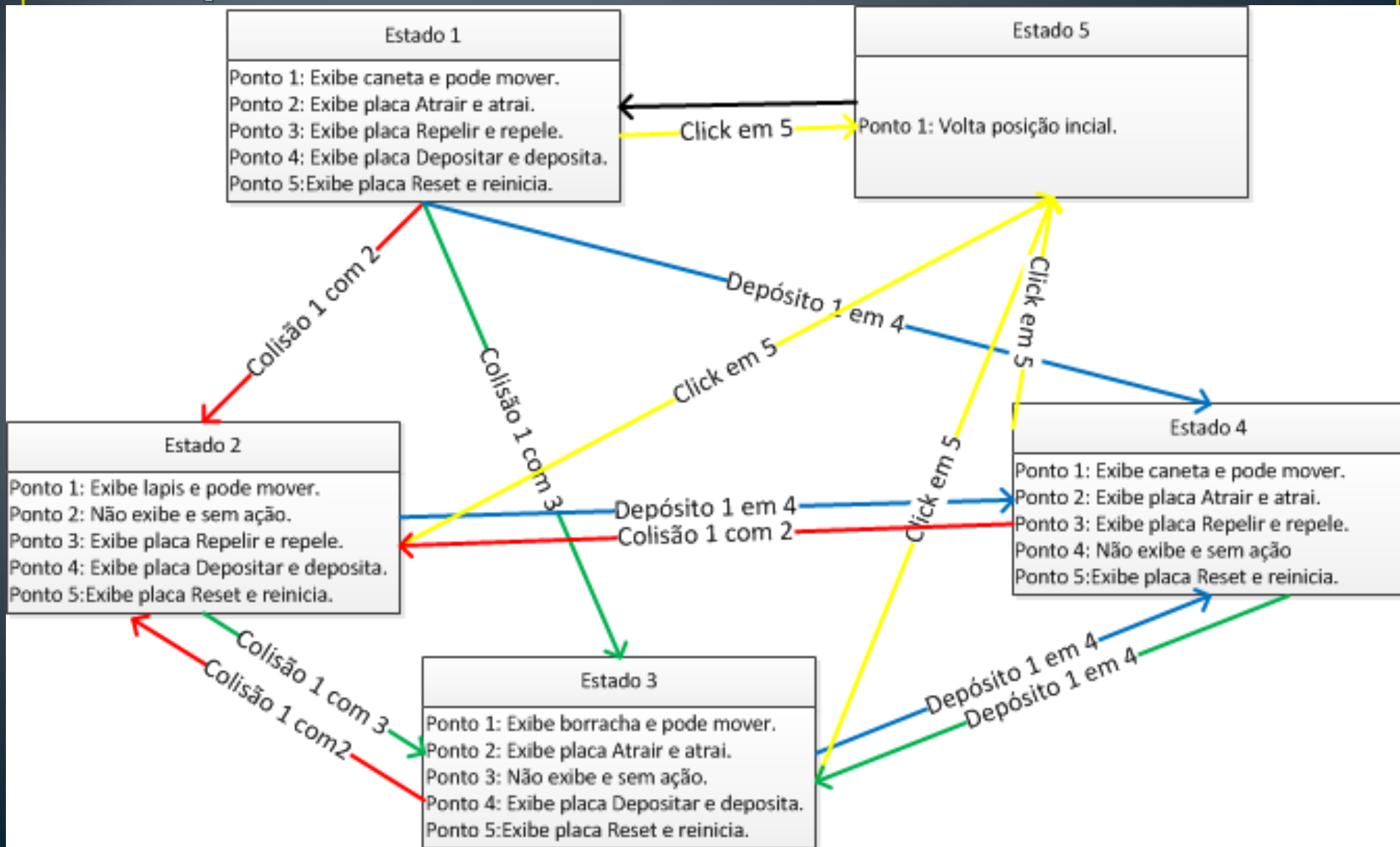
Exibe placa Depositar e deposita.

Exibe placa Reset e reinicia.

# Máquina de Estados

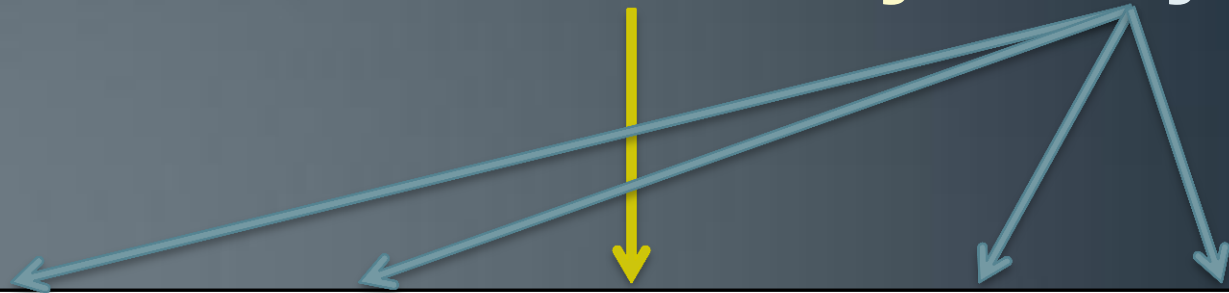
- 1 estado → Aplicação estática
- Vários estados → Aplicação dinâmica.
- A interligação destes estados: Máquina de estados, via sucesso de ações.

# Máquina de Estados



# Comportamento em comandos

- **Comportamento = Modo de exibição + Ação**



[ID\_PONTO] [OPCODE] <PARAMETER> [SHOWMODE] <NEXTSTATE> <AUDIO>

- Atrair apenas um ponto específico (Attract Only – ATTO)

Base ID	Point ID	ATTO	PointWaited	ShowMode	NextState	Audio	Overplay
---------	----------	------	-------------	----------	-----------	-------	----------

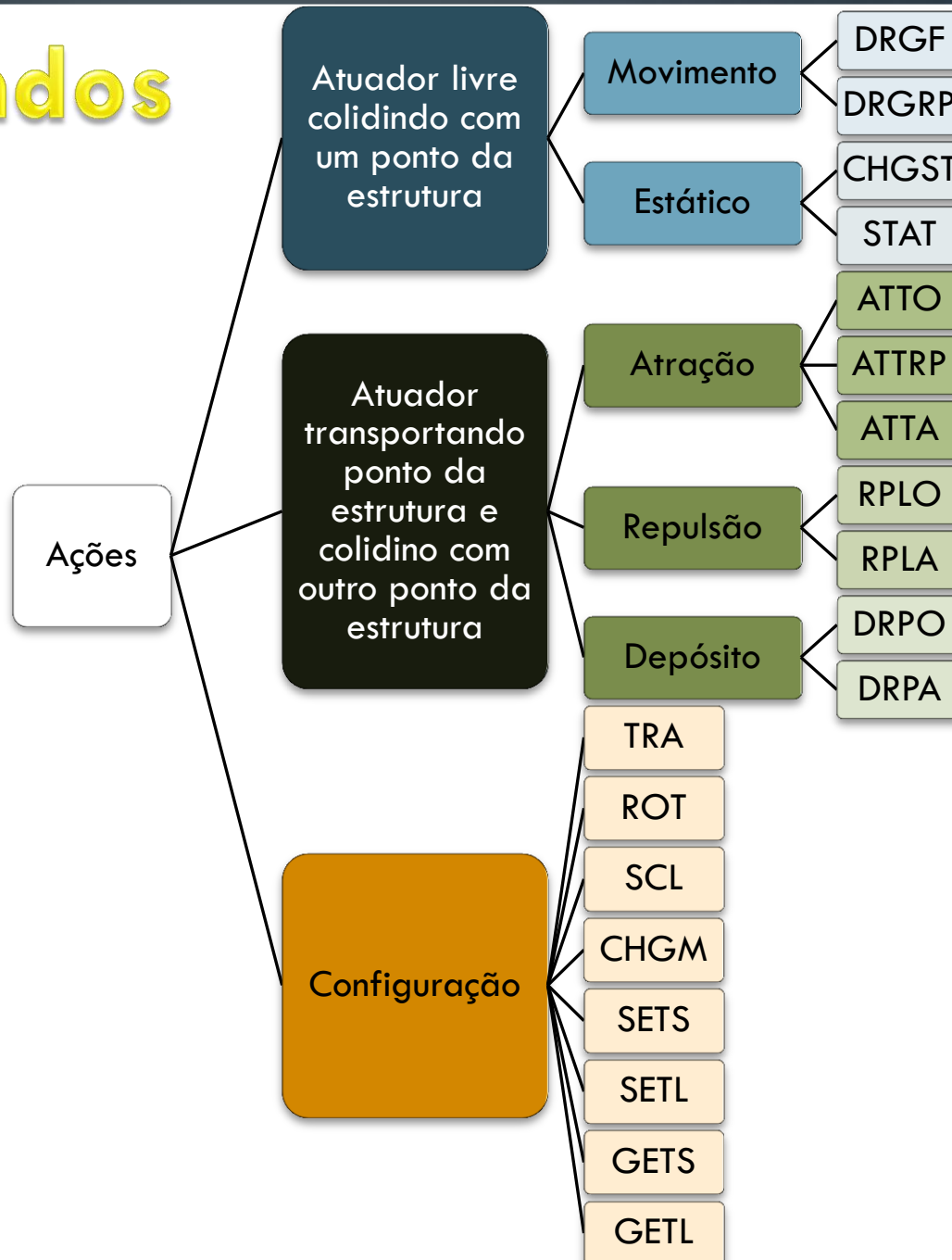
- Ponto estático (Static – STAT)

Base ID	Point ID	STAT	ShowMode	Audio	Overplay
---------	----------	------	----------	-------	----------

# Possibilidades de exibição do ponto de ação e dos modelos associados.

- 7 modos de visualização do ponto de ação no basAR:
  - **HIDE**: Tudo invisível.
  - **ONLY\_BALL**: Apenas a bola do ponto de ação é exibida.
  - **ONLY\_OBJECT**: Apenas o objeto do ponto de ação é exibido.
  - **BOTH**: A bola e o objeto ativo são exibidos.
  - **FLASH\_BALL**: Apenas a bola do ponto de ação é exibida piscando.
  - **SENSE\_PROX**: Apenas a bola do ponto de ação é exibida, com função de sensor de proximidade.
  - **ALL\_OBJECTS**: Mostra todos os objetos e a bola do ponto de ação.

# Comandos



# Comandos

- 19 comandos.
- 2 futuros para comandos externos.
- 1 futuro para exibir modelo em segunda tela.

## Addend 1 – Behavior Commands Set

### Commands:

1. Static, no action defined
[BaseID] [PointID] STAT [ShowMode] <AUDIO> <OVER?>
2. This action allows to drag the action point freely
[BaseID] [PointID] DRGF [ShowMode] <AUDIO> <OVER?>
3. This action allows to drag the action point freely and return to origin if collided
[BaseID] [PointID] DRGRP [ShowMode] [NextState] <AUDIO> <OVER?>
4. This action attracts one specific transporting action point
[BaseID] [PointID] ATTO [PointWaited] [ShowMode] [NextState] <AUDIO> <OVER?>
5. This action attracts one specific transporting action point and repels others
[BaseID] [PointID] ATTRP [PointWaited] [ShowMode] [NextState] <AUDIO> <OVER?>
6. This action attracts all action points
[BaseID] [PointID] ATTA [ShowMode] [NextState] <AUDIO> <OVER?>
7. This action allows to drop one specific transporting action point
[BaseID] [PointID] DRPO [PointWaited] [ShowMode] [NextState] <AUDIO> <OVER?>
8. This action allows to drop all action points
[BaseID] [PointID] DRPA [ShowMode] [NextState] <AUDIO> <OVER?>
9. This action repels one specific transporting action point
[BaseID] [PointID] RPLO [PointWaited] [ShowMode] [NextState] <AUDIO> <OVER?>
10. This action repels all action points
[BaseID] [PointID] RPLA [ShowMode] [NextState] <AUDIO> <OVER?>
11. This action changes the actual state to a next state
[BaseID] [PointID] CHGST [ShowMode] [NextState] <AUDIO> <OVER?>
12. This action translates the action point
[BaseID] [PointID] TRA [X] [Y] [Z]
13. This action rotates the action point
[BaseID] [PointID] ROT [X] [Y] [Z]
14. This action scales the action point
[BaseID] [PointID] SCL [X] [Y] [Z]
15. This action changes the active model of the action point
[BaseID] [PointID] CHGM [ModelToChange]
16. This action saves Actual position on Start
[BaseID] [PointID] SETS
17. This action saves Actual position on Last
[BaseID] [PointID] SETL
18. This action saves Start position on Actual
[BaseID] [PointID] GETS
19. This action saves Last position on Actual
[BaseID] [PointID] GETL

[ShowMode] - Point visibility mode

[PointWaited] - Object expect to the action

[NextState] - Next state if action is accomplished

[ModelToChange] - Model to change on CHGM command

[X], [Y], [Z] - Angle coordinates to

<AUDIO> - Sound to play if action is accomplished (May not be used). DRGRP, ATTO, ATTRP, DRPO generate error sound.

<OVER?> - Stops any other sound and play only this one.

# Tipos de estados

## Estado de Trabalho

- Transição via sucesso de ações.
- END\_STATE

BEGIN\_STATE 3

Comandos.

END\_STATE

## Estado de Configuração

- Transição automática.
- END\_STATE **GO\_TO <NS>**

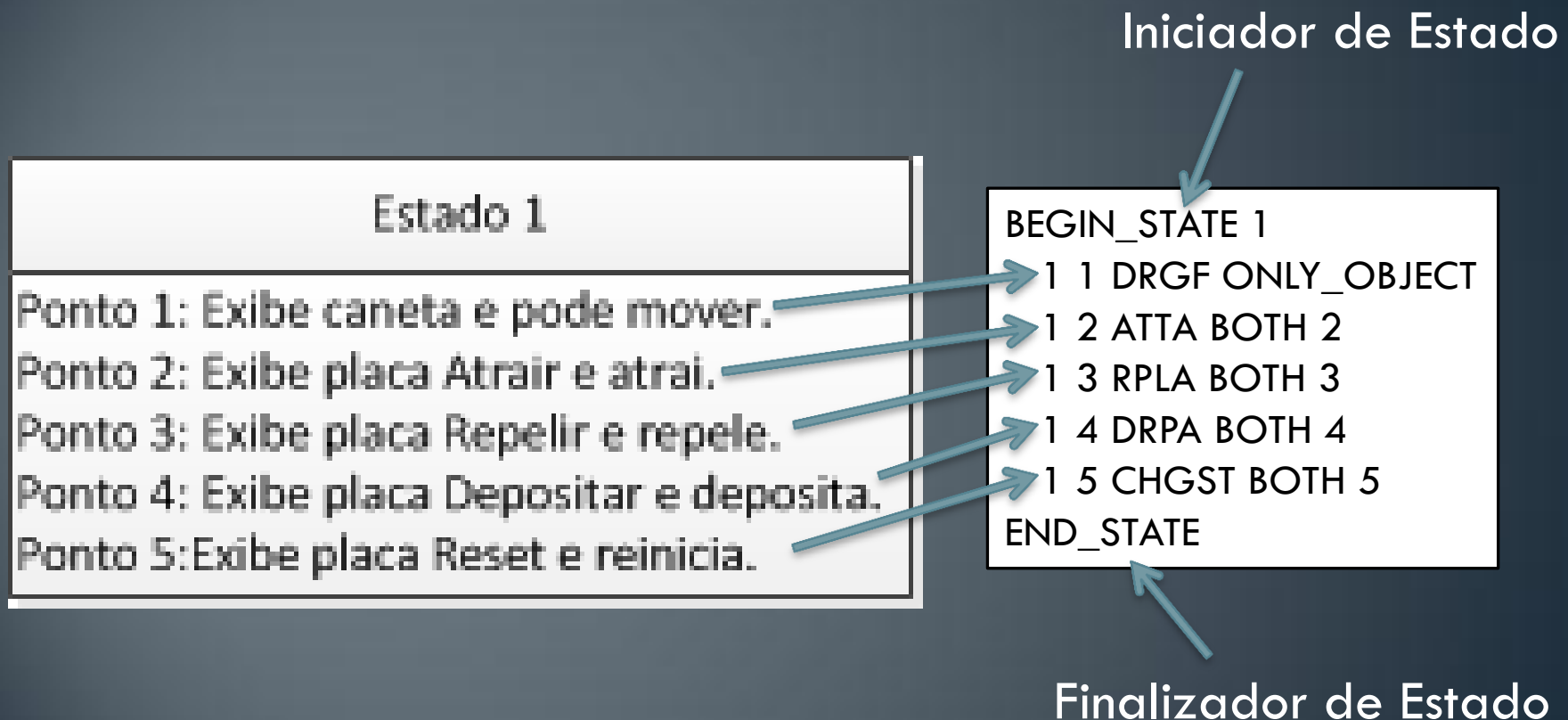
BEGIN\_STATE 3

Comandos, com comandos de configuração.

END\_STATE GO\_TO 4



# Estado em comandos



BEGIN\_STATE 1

1 1 DRGF ONLY\_OBJECT

1 2 ATTA BOTH 2

1 3 RPLA BOTH 3

1 4 DRPA BOTH 4

1 5 CHGST BOTH 5

END\_STATE

Estado 5

BEGIN\_STATE 5

1 1 GETS

END\_STATE GO\_TO 1

BEGIN\_STATE 2

1 1 DRGF ONLY\_OBJECT

1 2 STAT HIDE

1 3 RPLA BOTH 3

1 4 DRPA BOTH 4

1 5 CHGST BOTH 5

END\_STATE

BEGIN\_STATE 3

1 1 DRGF ONLY\_OBJECT

1 2 ATTA BOTH 2

1 3 STAT HIDE

1 4 DRPA BOTH 4

1 5 CHGST BOTH 5

END\_STATE

BEGIN\_STATE 4

1 1 DRGF ONLY\_OBJECT

1 2 ATTA BOTH 2

1 3 RPLA BOTH 3

1 4 STAT HIDE

1 5 CHGST BOTH 5

END\_STATE

Click em 5

Depósito 1 em 4

Click em 5

Colisão 1 com 3

Depósito 1 em 4

Colisão 1 com 2

Click em 5

Depósito 1 em 4

Depósito 1 em 4

Colisão 1 com 3

Colisão 1 com 2

# Configurando comportamento

- No basAR um arquivo de mantém todo o comportamento do aplicativo.
- O arquivo que configura toda a aplicação chama-se data/**config\_behavior** e este é indicado no arquivo data/config\_basar.
- **No primeiro estado tem que ser configurado todos os pontos.**
- **Se não modificar a condição do ponto, é mantido a do estado anterior.**
- **Mesmo se a aplicação for estática, deve conter o primeiro estado.**

# Exemplo: config\_behavior

```
BEGIN_STATE 1
  1 1 DRGF ONLY_OBJECT
  1 2 ATTA BOTH 2
  1 3 RPLA BOTH 3
  1 4 DRPA BOTH 4
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 2
  1 1 DRGF ONLY_OBJECT
  1 2 STAT HIDE
  1 3 RPLA BOTH 3
  1 4 DRPA BOTH 4
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 3
  1 1 DRGF ONLY_OBJECT
  1 2 ATTA BOTH 2
  1 3 STAT HIDE
  1 4 DRPA BOTH 4
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 4
  1 1 DRGF ONLY_OBJECT
  1 2 ATTA BOTH 2
  1 3 RPLA BOTH 3
  1 4 STAT HIDE
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 5
  1 1 GETS
END_STATE GO_TO 1
```

# Até agora tudo bem?

# Executando exemplo

<https://sites.google.com/site/christophercerqueira/projetos/ear/basar>

- Mudar posição
- Mudar ordem dos modelos
- Mudar função dos pontos.

Alterações no  
exemplo

# Mudar posição dos pontos

- Abrir arquivo data/config\_base

...

5

Objeto

DEFAULT\_IPOINT

Data/app\_objeto

**200.0 0.0 0.0**

0.0 0.0 0.0

1 1 1

500.0

Atracao

DEFAULT\_IPOINT

Data/app\_at

**100.0 -100.0 0.0**

0.0 0.0 0.0

1 1 1

500.0

Repulsao

DEFAULT\_IPOINT

Data/app\_rp

**100.0 0.0 0.0**

0.0 0.0 0.0

1 1 1

500.0

Deposito

DEFAULT\_IPOINT

Data/app\_dp

**100.0 100.0 0.0**

0.0 0.0 0.0

1 1 1

500.0

Reset

DEFAULT\_IPOINT

Data/app\_rs

**100.0 200.0 0.0**

1 1 1

500.0



# Mudar ordem dos modelos

- Abrir o arquivo data/config\_object

```
#Data/app_object
```

```
3
```

```
MODEL3D VRML Wrl/caneta.dat
```

```
MODEL3D VRML Wrl/borracha.dat
```

```
MODEL3D VRML Wrl/lapis.dat
```

# Mudar função dos pontos

- Abrir o arquivo data/config\_behavior par modificar o comportamento do ponto.
- Ponto 2 → Repelir
- Ponto 3 → Depósito
- Ponto 4 → Atrair

# Parte 1 mudar no comportamento

```
BEGIN_STATE 1
  1 1 DRGF ONLY_OBJECT
  1 2 RPLA BOTH 2
  1 3 DRPA BOTH 3
  1 4 ATTA BOTH 4
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 2
  1 1 DRGF ONLY_OBJECT
  1 2 STAT HIDE
  1 3 DRPA BOTH 3
  1 4 ATTA BOTH 4
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 3
  1 1 DRGF ONLY_OBJECT
  1 2 RPLA BOTH 2
  1 3 STAT HIDE
  1 4 ATTA BOTH 4
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 4
  1 1 DRGF ONLY_OBJECT
  1 2 RPLA BOTH 2
  1 3 DRPA BOTH 3
  1 4 STAT HIDE
  1 5 CHGST BOTH 5
END_STATE
```

```
BEGIN_STATE 5
  1 1 GETS
END_STATE GO_TO 1
```

ATTA  
RPLA  
DRPA

## Parte 2 mudar as placas dos pontos

...

5

Objeto

DEFAULT\_IPOINT

Data/app\_objeto

200.0 0.0 0.0

0.0 0.0 0.0

1 1 1

500.0

**Repulsao**

DEFAULT\_IPOINT

Data/app\_rp

100.0 -100.0 0.0

0.0 0.0 0.0

1 1 1

500.0

**Deposito**

DEFAULT\_IPOINT

Data/app\_dp

100.0 0.0 0.0

0.0 0.0 0.0

1 1 1

500.0

**Atracao**

DEFAULT\_IPOINT

Data/app\_at

100.0 100.0 0.0

0.0 0.0 0.0

1 1 1

500.0

Reset

DEFAULT\_IPOINT

Data/app\_rs

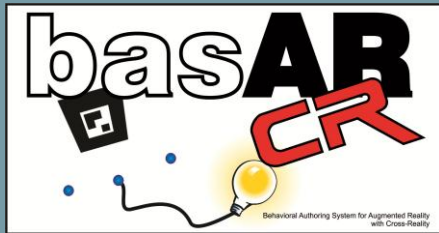
100.0 200.0 0.0

1 1 1

500.0

<https://sites.google.com/site/christophercerqueira/projetos/ear/basar>

## Conclusões



<https://sites.google.com/site/christophercerqueira/projetos/ear/basar>

- Mais poderoso e mais complexo.
- Potencial da interatividade.
- Ineditismo nos trabalhos.
- Parte do contexto do projeto ATIRA.

## • PRÓXIMAS VERSÕES

- Novos comandos
  - ARDUINO e SEGUNDA TELA
- Atuador com WI/KINECT
- Outros formatos de arquivos 3D
- Interação entre bases.
- Interface gráfica

# Aproveitando o público

- Testar aplicação matemática.
- Responder questionário.