



ARDUINO

Curso basAR

Christopher Shneider Cerqueira – christophercerqueira@gmail.com

Claudio Kirner – ckirner@gmail.com

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

<http://www.ckirner.com/basar>

Requisitos

- Windows
- basAR
 - Download:
 - <https://sites.google.com/site/christophercerqueira/projetos/ear/basar>
 - <http://www.ckirner.com/basar>
- Alguns modelos 3D em VRML
- ARDUINO
- Editor de texto simples, como o Bloco de Notas.

Roteiro

- Introdução: Metas
- basAR
- Criando infraestrutura, estrutura e conteúdo
- Criando comunicação com o ARDUINO
 - Conceitos:
 - Tabela de comandos
 - Envio de comandos
 - Recepção de comandos
 - Interrupção
 - Comandos
 - Alterando comandos
 - Atividade

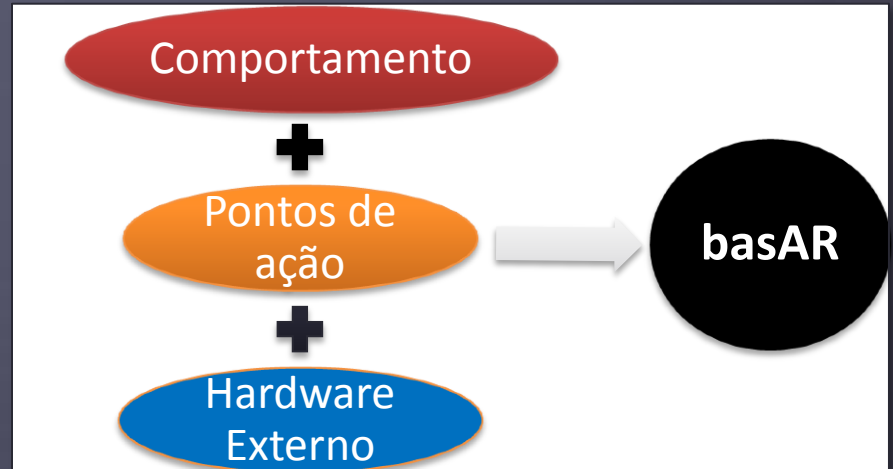
Introdução

- Criar um cenário de Realidade Aumentada.
 - Dois pontos de seleção e um ponto de exibição
- Utilizar o software basAR.
- Entender a construção do cenário.
- Entender os conceitos de mudança de estado e mudança de atributos

basAR

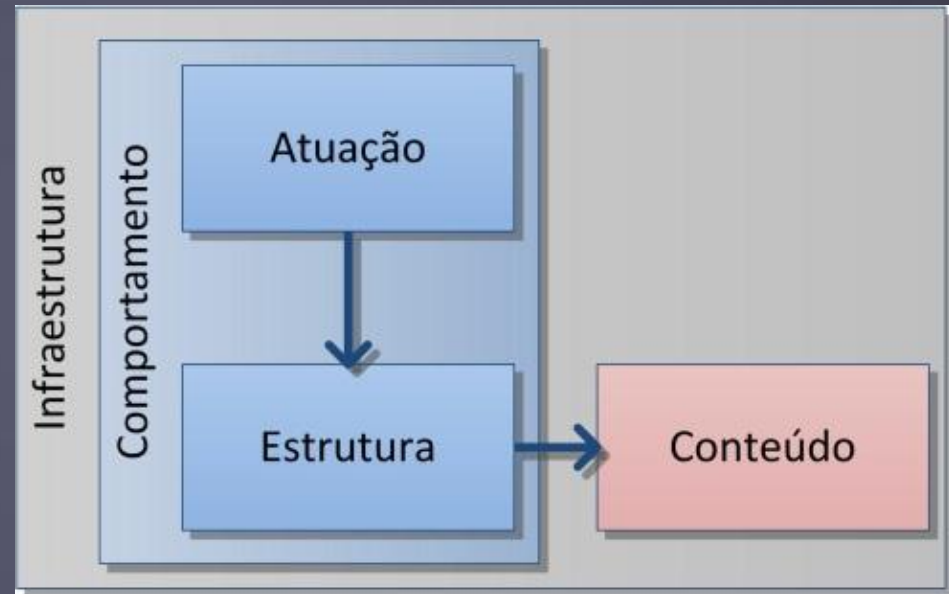
basAR: Behavioral Authoring System for Augmented Reality

- Meta-ferramenta de autoria de ambientes inteligentes de realidade aumentada .
 - Estrutura de comportamento baseada em pontos de ação
 - Cross-Reality
- COMPORTAMENTO DINÂMICO entre PONTOS DE AÇÃO

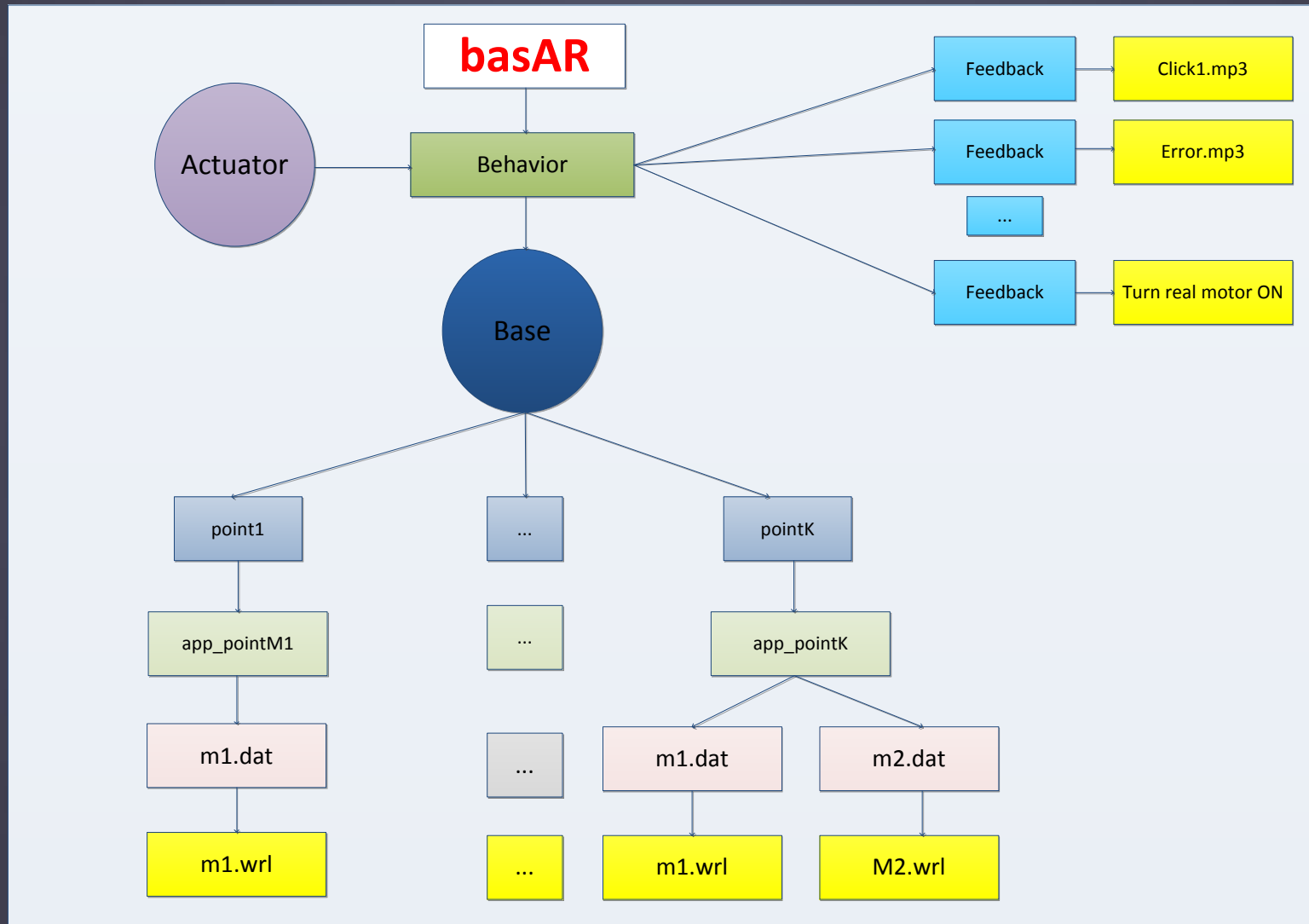


Aplicação em camadas

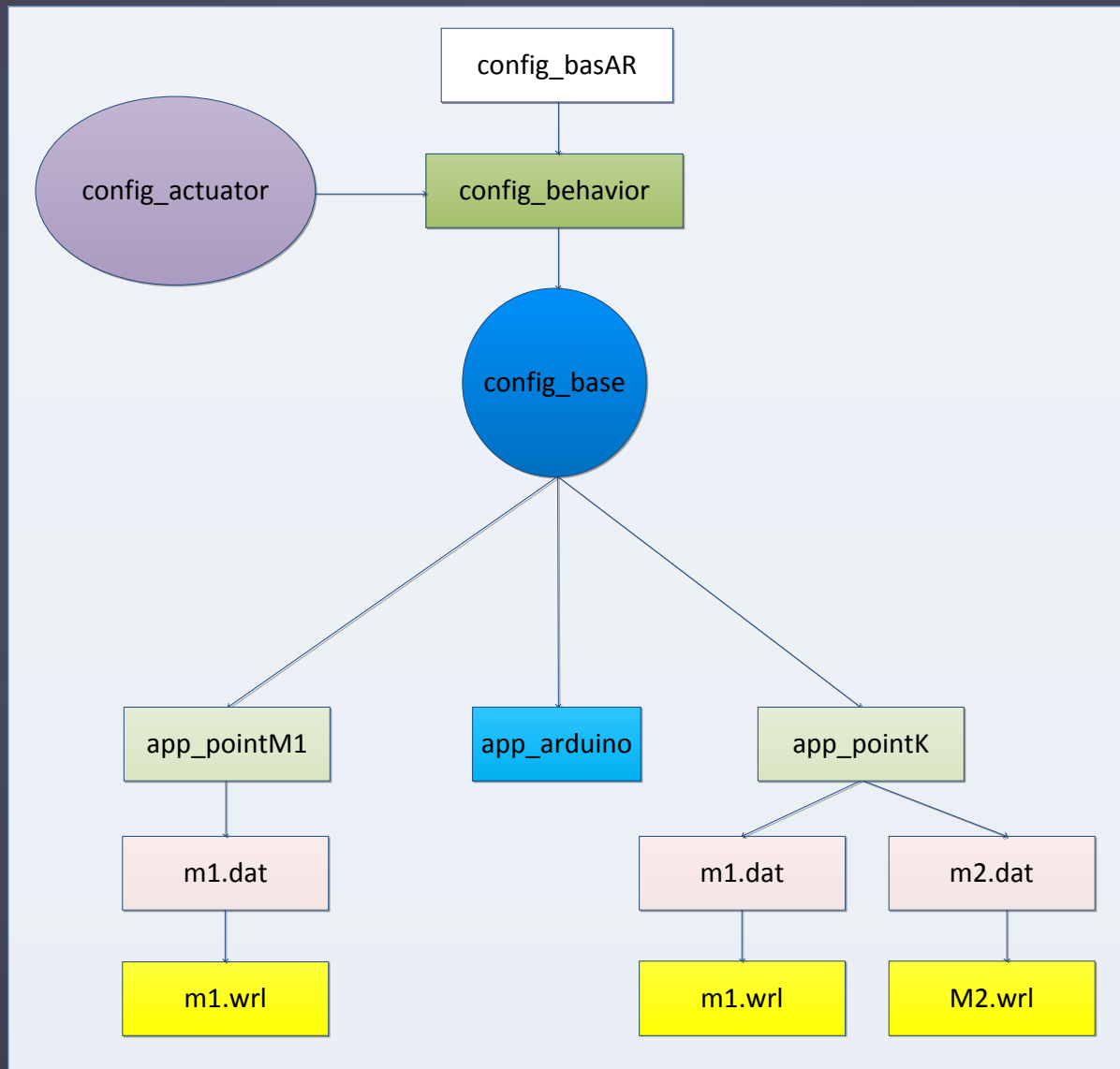
- **Infraestrutura:** especificar a área de trabalho (*workspace*) da aplicação
- **Estrutura:** os posicionamentos dos pontos de ação sobre a infraestrutura.
- **Contexto:** modelos, sons e outros objetos definidos na estrutura.
- **Atuação:** método de interação com a estrutura.
- **Comportamento:** regras de interação entre os atuadores e a estrutura.



Hierarquia de objetos



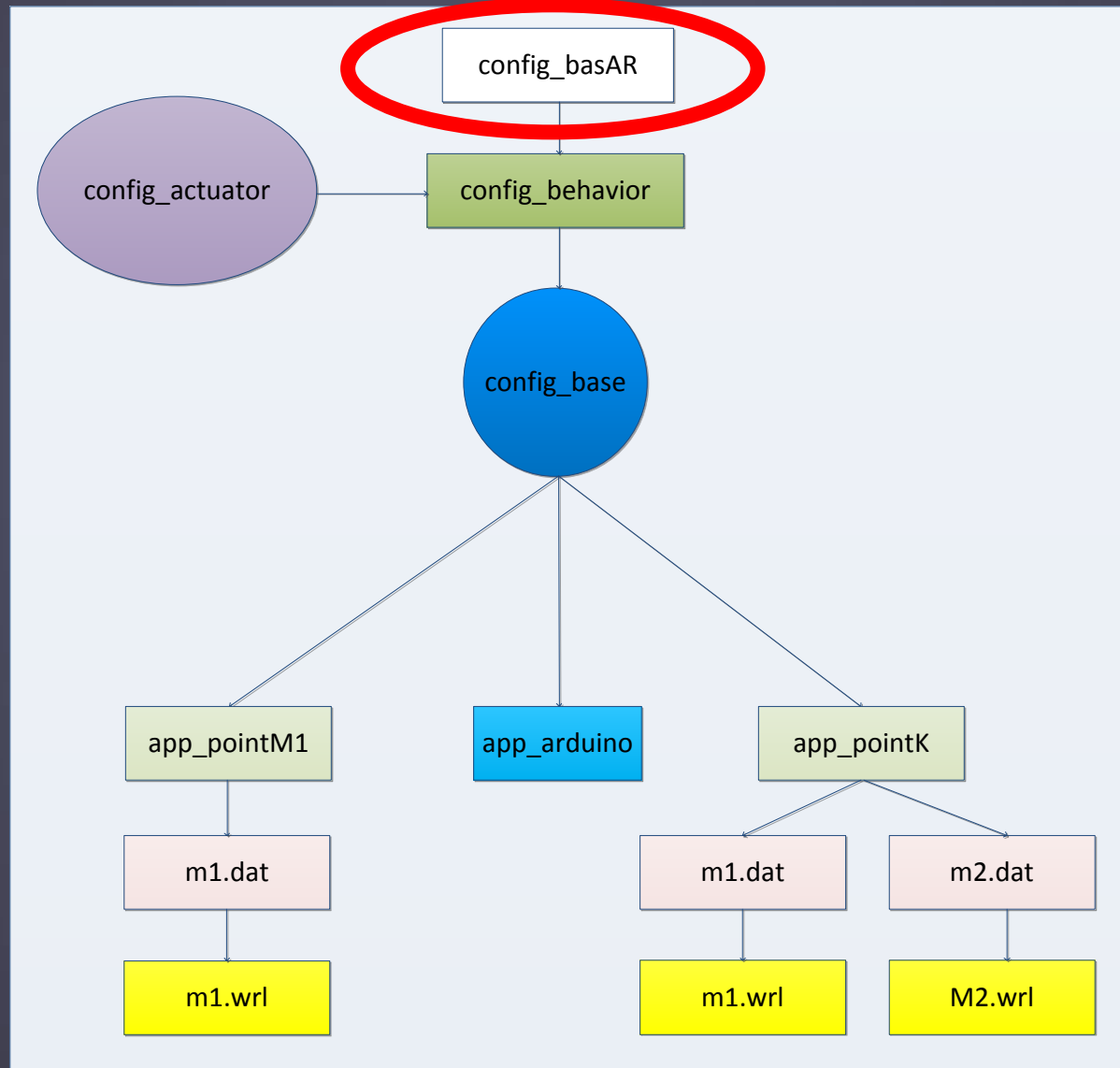
Hierarquia de arquivos



Criando

INFRAESTRUTURA, ESTRUTURA, CONTEÚDO E ATUAÇÃO

Sistema



Sistema

- Configura:
 - a primeira interligação dos objetos;
 - modo de exibição da janela;
 - objetos padrões.
- Arquivo:
{basAR}/Data/config_basAR

#config_basAR

basAR

WINDOWED

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 **# Rule Machine**

Bases

1

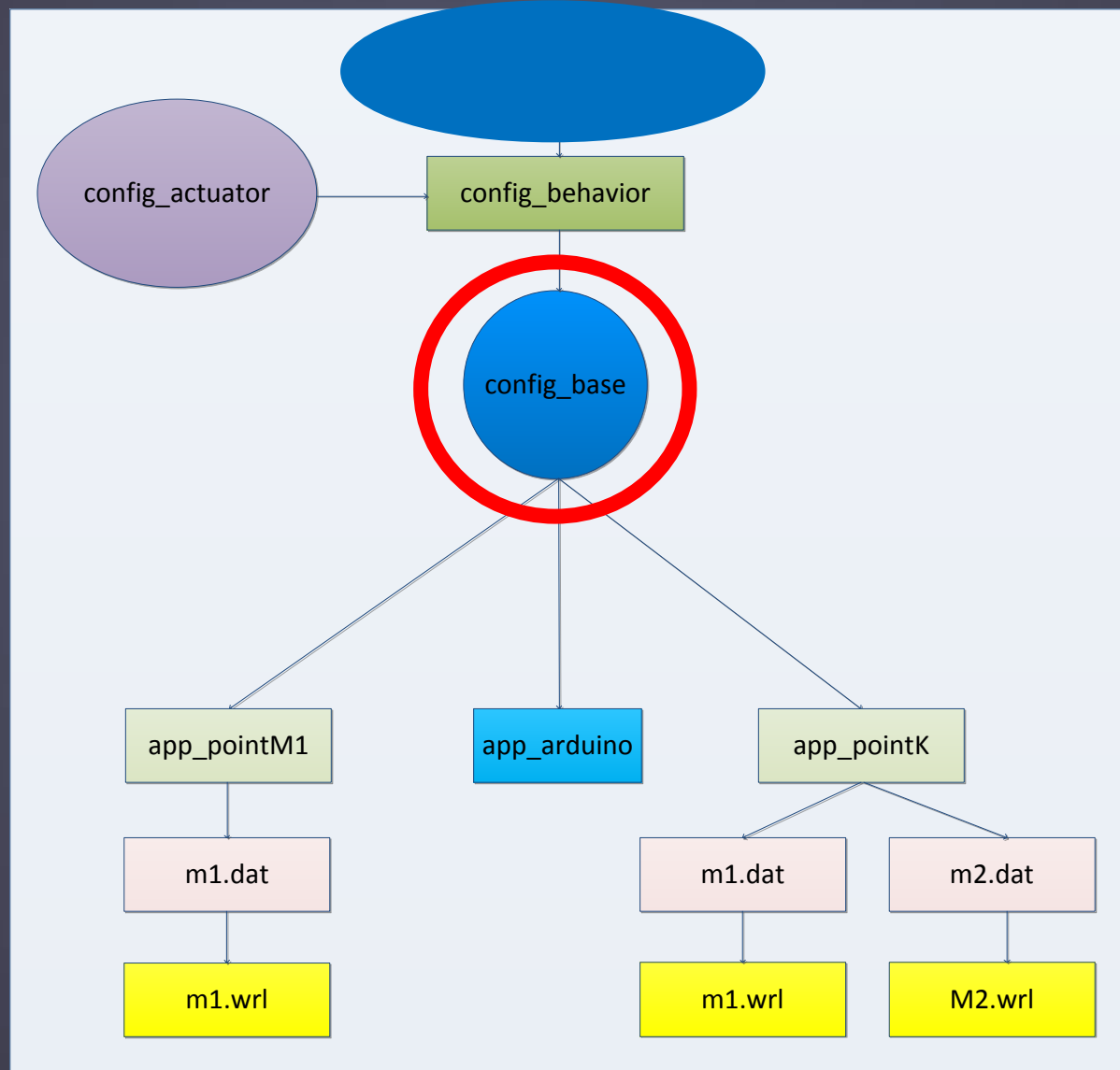
Data/config_base1

Actuators

1

ARTKSM Data/config_transport

Infraestrutura



Infraestrutura

- correlação entre o mundo real e o mundo virtual
- Definição de base
- Arquivo:
{basAR}/Data/config_base

```
#config_base1
```

```
# This file contains the setup for a workspace  
BASE1
```

```
# Single ARToolKit Marker configuration
```

```
ARTKSM
```

```
Data/Markers/base.patt
```

```
53.0
```

```
0.0 0.0
```

```
USE_DEFAULT
```

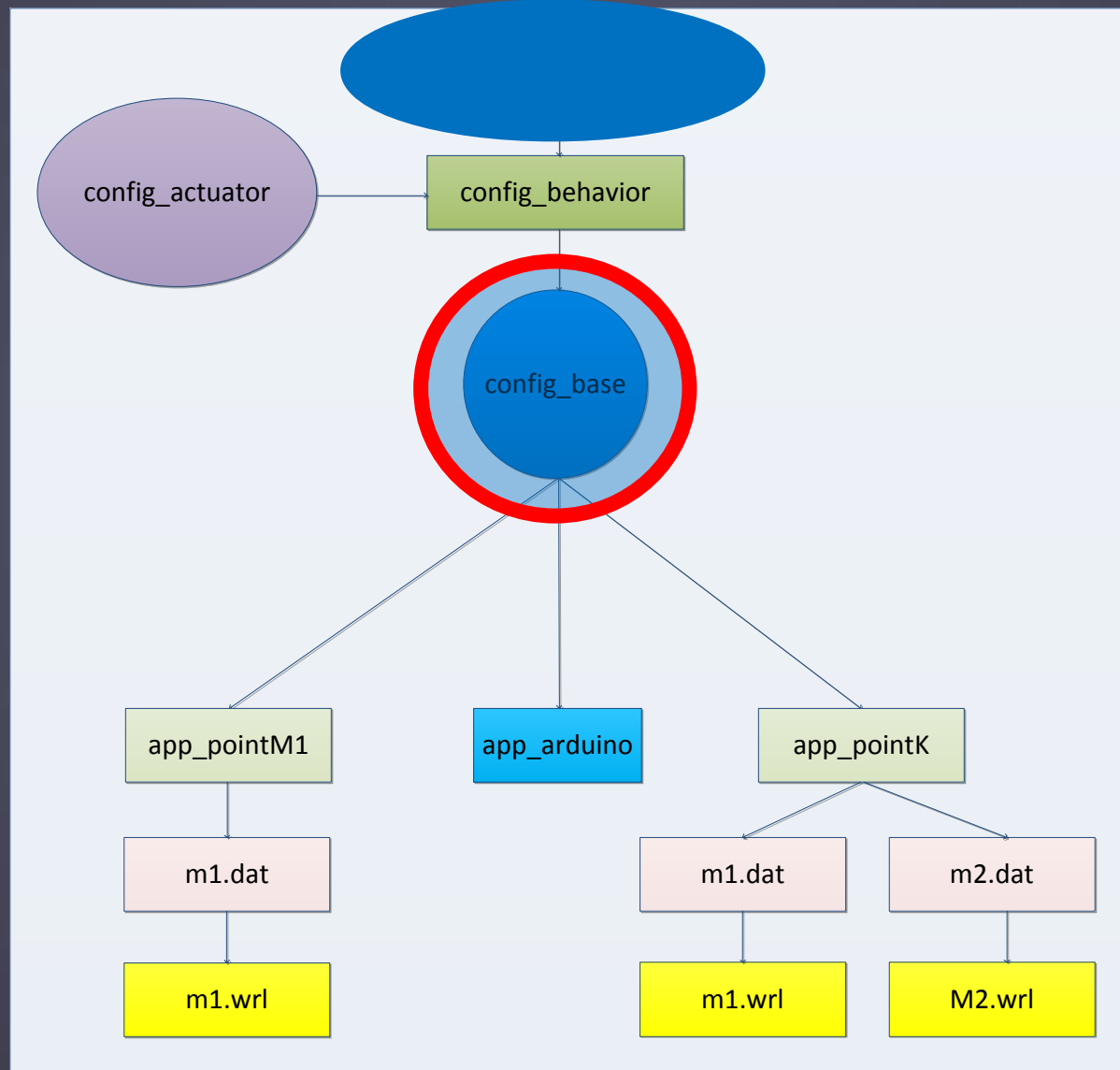
```
# Workspace Sounds
```

```
Audio/bell.wav    ONCE 0.5    # Visible Sound
```

```
Audio/explosion.wav 0.5      # Error sound
```

```
wrl/action/status.dat
```

Estrutura



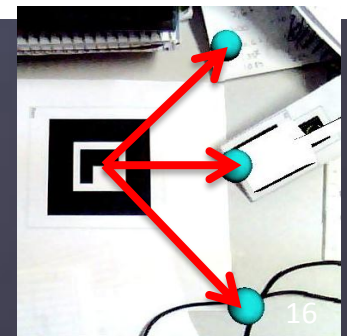
Estrutura

- Pontos do espaço virtual com **esferas de ação**, **modelos associados** e **comportamento**.
- Localização, orientação e escala dos objetos virtuais.
- Arquivo:
{basAR}/Data/config_base

...

Pen	# Point Name
DEFAULT_IPOINT	# Action Model File
Data/app_pen	# OBJECT Model File
20.0 20.0 0.0	# Translation
0.0 0.0 0.0	# Rotation
1.0 1.0 1.0	# Scale
900.0	# Action radius

...



Exemplo com 3 pontos

...

2

Ponto

DEFAULT_IPOINT

Data/App/app_object

100.0 100.0 25.0

0.0 0.0 0.0

1.0 1.0 1.0

300.0

Arduino

EXTERN_IPOINT

Data/App/app_arduino

100.0 0.0 25.0

0.0 0.0 0.0

1.0 1.0 1.0

0

{basAR}/Data/config_base

Arquivo config_base completo _p1

BASE1

ARTKSM

Data/Markers/base.patt

53.0

0.0 0.0

USE_DEFAULT

Audio/bell.wav ONCE 0.5

Audio/explosion.wav 0.5

wrl/action/status.dat

2

Ponto

DEFAULT_IPOINT

Data/App/app_object

100.0 100.0 25.0

0.0 0.0 0.0

1.0 1.0 1.0

300.0

Arquivo config_base completo _p2

Arduino

EXTERN_IPOINT

Data/App/app_arduino

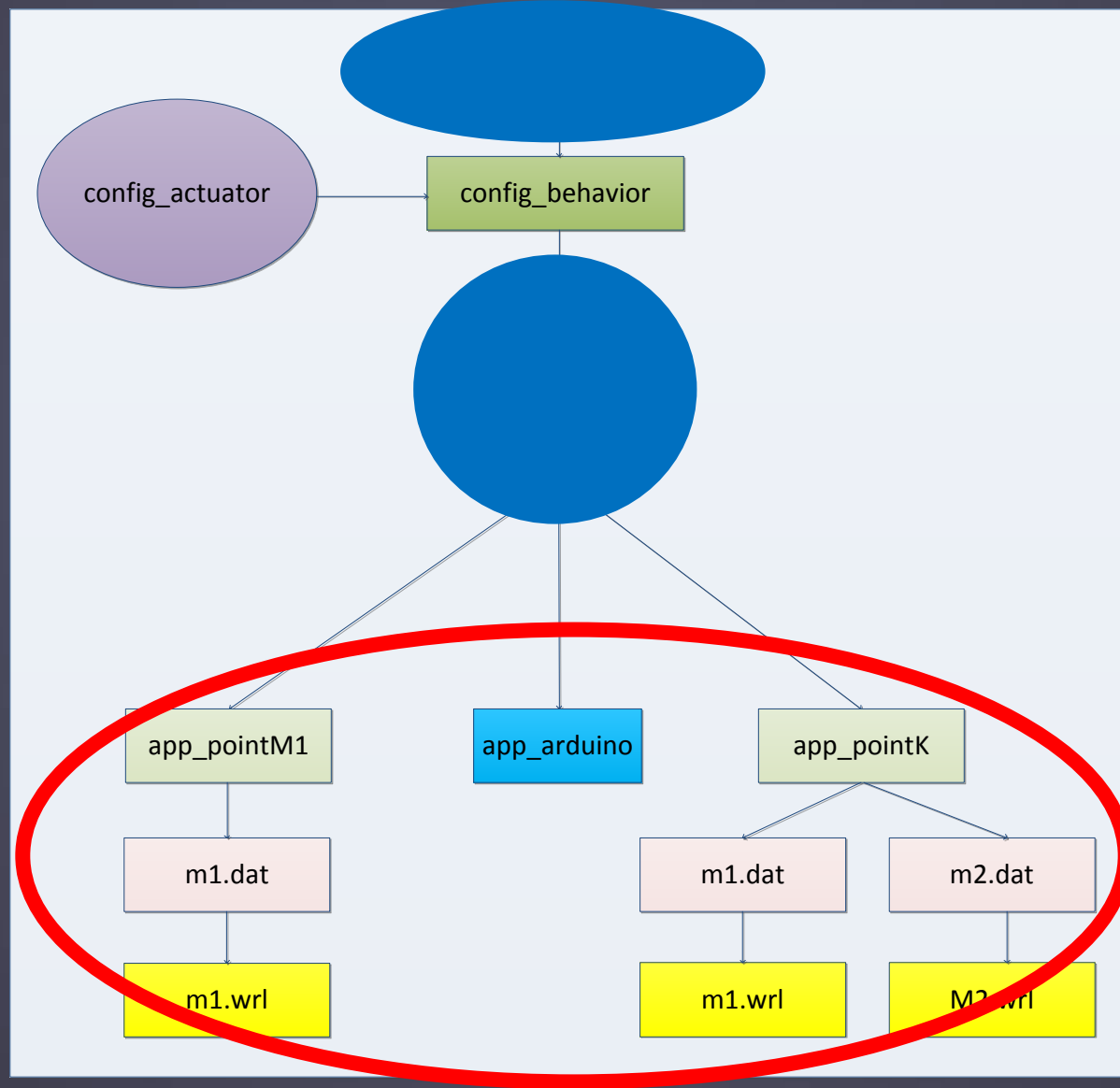
100.0 0.0 25.0

0.0 0.0 0.0

1.0 1.0 1.0

0

Conteúdo



Conteúdo

- objetos 3D e áudios que são utilizados.

- Arquivos:

{basAR}/Data/App/app_obj

{basAR}/Wrl/quad.dat

{basAR}/Wrl/triang.dat

{basAR}/Wrl/quad.wrl

{basAR}/Wrl/triang.wrl

#app_obj

2

Number of objects

MODEL3D VRML Wrl/quad.dat

MODEL3D VRML Wrl/triang.dat

#quad.dat

quad.wrl

0 0 0

0 90 0

0 0 0

triang.dat

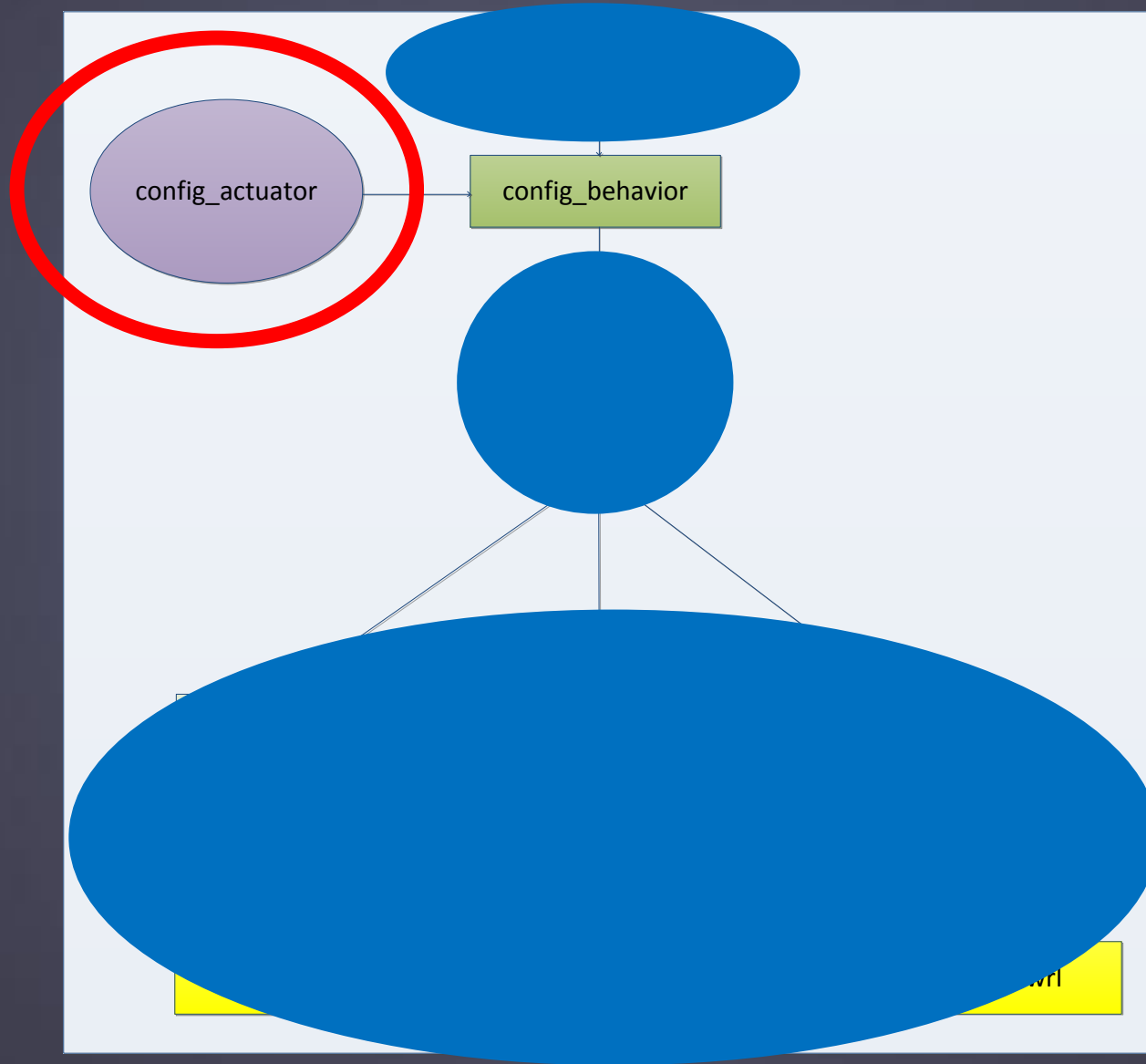
triang.wrl

0 0 0

0 90 0

0 0 0

Atuação



Atuação

- Interação do usuário com o sistema



#config_transport

ARTKSM1

Single ARToolKit Marker configuration

Data/Markers/shovell.patt

37.0

0.0 0.0

USE_DEFAULT

VRML wrl/Action/ssd.dat

model

Marker

Width(mm)

Central

Marker cover

Symbolic

Collision point

DEFAULT_IPOINT

20.0 0.0 0.0

400.0

Point model

Translation (x,y,z) (mm)

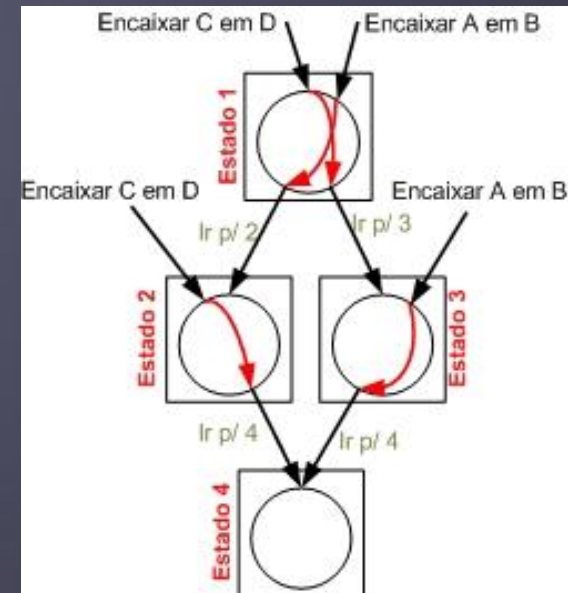
Action radius of the point

Criando

COMUNICAÇÃO COM O ARDUINO

Comportamento?

- Controle do feedback do sistema ao receber estímulo do usuário.
- Baseado em máquina de estados (grafo de cena).
- Ações → Comandos
- Estados → Blocos de comandos



Exemplo de estado

- Inicia estado
- Ponto 1 estático, exibe tudo
- Ponto 1 estático, exibe tudo
- Ponto 1 estático, exibe tudo
- Finaliza estado

BEGIN_STATE 1

1 STAT BOTH

2 STAT BOTH

3 STAT BOTH

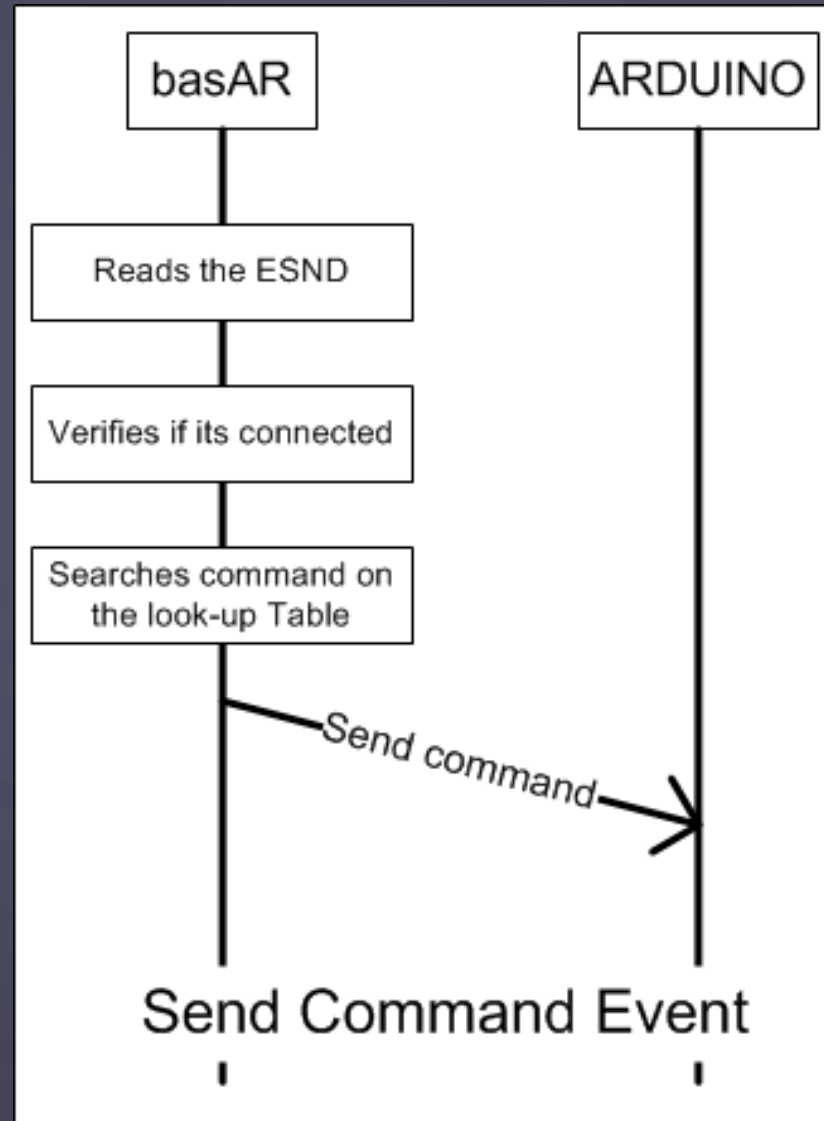
END_STATE

- Outros comandos vide manual do basAR!

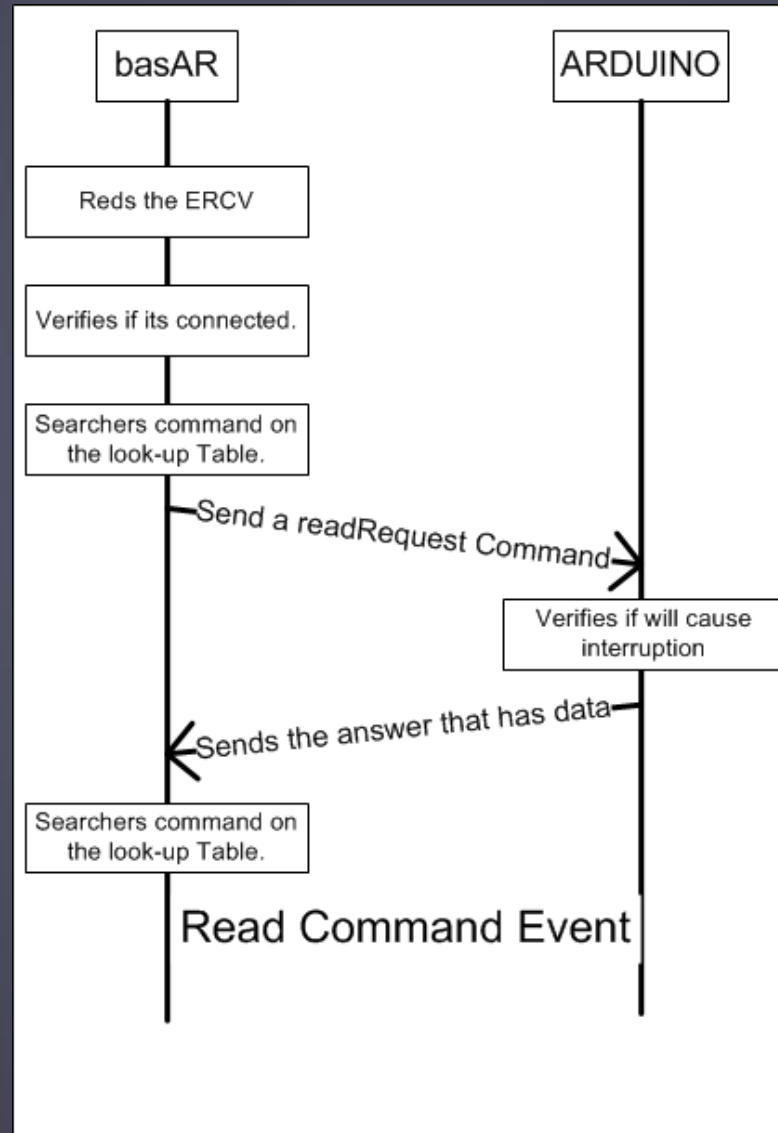
O que é Comunicação?

- Envio e recepção de dados entre dois ou mais sistemas.
- Protocolo de comunicação.
- Lookup table

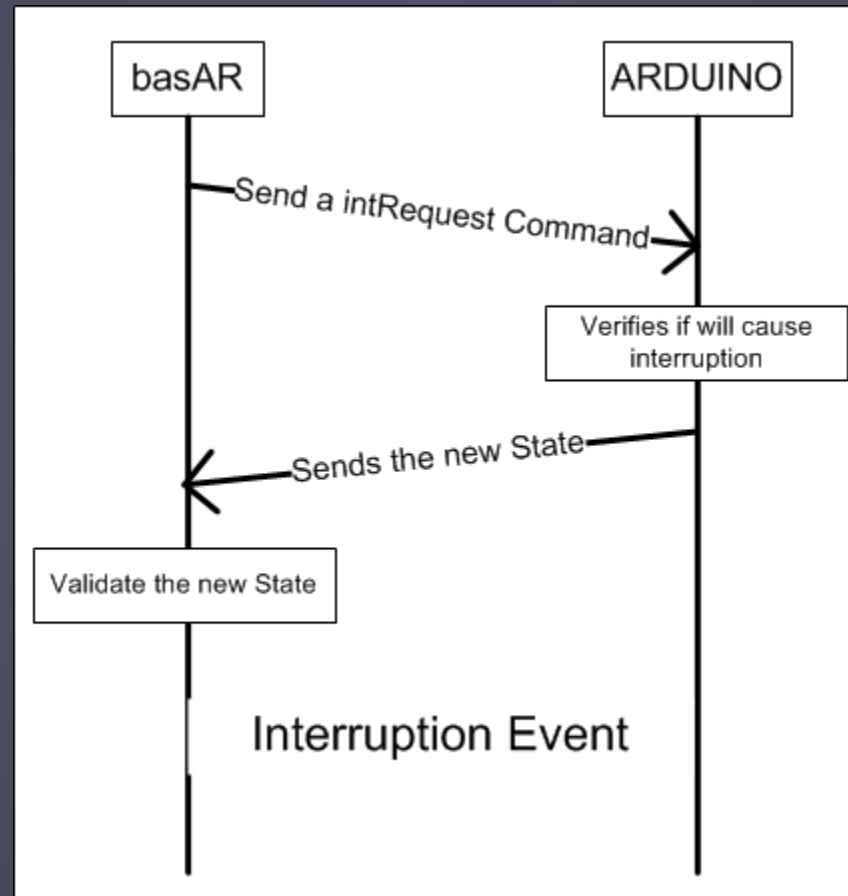
Evento de envio



Evento de Recepção



Evento de Interrupção



Lookup Table

- Tabela de tradução de comandos para enviar para o ARDUINO.
- Auxilia na separação de qual comando vai ser enviado para o ARDUINO

Lookup Table

COM4 # ARDUINO COM

ARDUINO LOOKUP TABLE

THE NUMBERS 00-09, and 255 are RESERVED to Configuration and Test. 10-254

TABLE FORMAT: REQUEST_NUMBER REQUEST_NAME <NEXT_STATE>

CONFIGURATION AND TEST COMMANDS

1 aliveTest

2 aliveAnswer

3 intRequest

4 readRequest

5 readRequestAnswer

USER COMMANDS

DEVICE 1 - Light

10 lightOFF

11 lightON

DEVICE 2 - Button

20 buttonPressed 5

ARDUINO CODE

```
// start global variables
void setup() {
  Serial.begin(9600);    // connect to the serial port
}

void loop () {
  // read the serial port
  val = Serial.read();
  // Code to adapt data
  switch(val){
  case 1: { // CHECK IF ITS aliveTest
    Serial.print(2);    // send back aliveAnswer
    break;  }
  case 3: { // CHECK IF ITS intRequest
    {Serial.print(intNSdata);
    break;  }
  case 4: { // CHECK IF Its readRequest
    break;  }
  default:
    break;  }
}
```

Comando de envio de mensagens

- Enviar mensagem (Send message)
 - ESND



- 1 ESND lightON

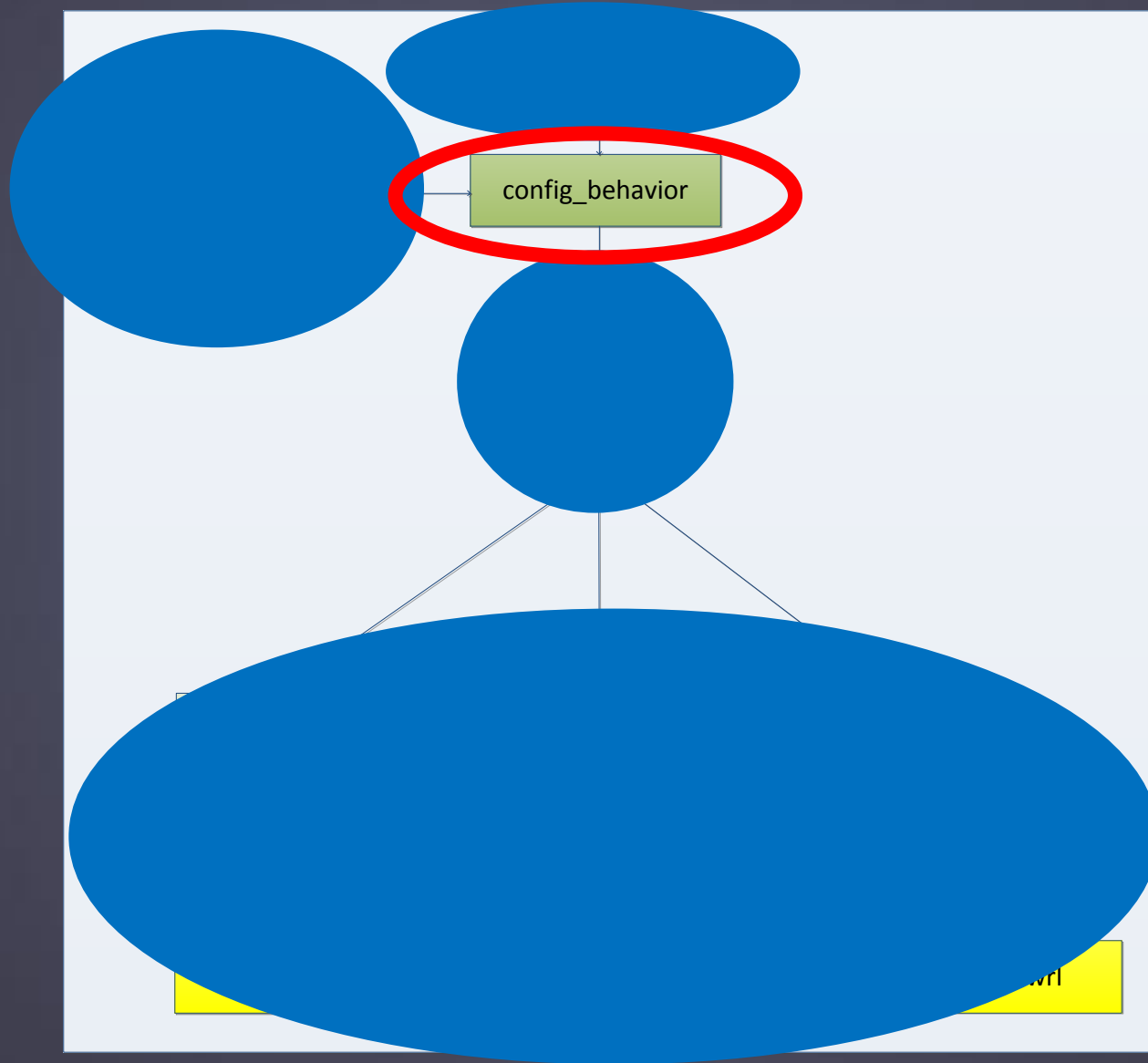
Comando de recebimento de mensagens

- Receber mensagem (Receive message)
 - ESND



- 1 ERCV buttonPressed

Comportamento



Arquivo de Comportamento

```
BEGIN_STATE 1
    1 CHGST ONLY_BALL 2
    2 ESND lightOFF
END_STATE AFTER 2
```

```
BEGIN_STATE 2
    1 CHGST ONLY_BALL 1
    2 ESND lightON
END_STATE AFTER 2
```

RESUMO

Resumo comandos utilizados

- CHGST**: [PointID] **CHGST** [ShowMode] [NextState] <AUDIO> <OVER?>
- ESND**: [PointID] **ESND** [Message]
- ERCV**: [PointID] **ERCV** [Message]

Ajuda online

- Videos exemplificando cada comportamento disponível e os arquivos utilizados neste tutorial estão disponíveis em:

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



Clicks

Curso basAR

Christopher Shneider Cerqueira – christophercerqueira@gmail.com

Claudio Kirner – ckirner@gmail.com

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

<http://www.ckirner.com/basar>