

# Agentes inteligentes y sistemas multiagentes

## Taller de Sistemas Multiagentes

Prof. Dr. Ariel Monteserin  
amontese@exa.unicen.edu.ar

ISISTAN – Fac. de Ciencias Exactas- UNICEN

CONICET

2017

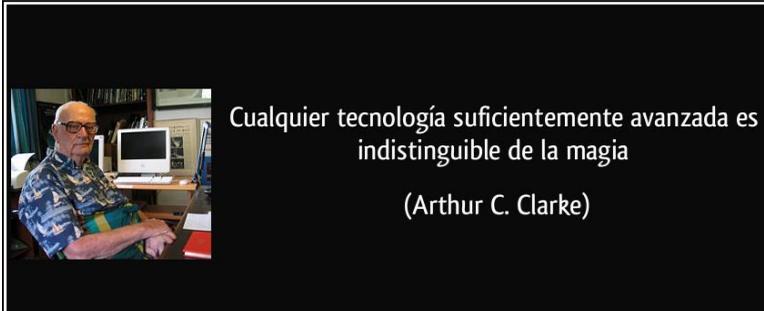
## Agenda

- La materia
- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitecturas abstractas y concretas
  - Agentes personales – móviles
- Sistemas multiagentes
  - Conceptos
  - Comunicación



## Taller de sistemas multiagentes

- Objetivo de la materia



- Cursada intensiva (traer notebooks)
- Trabajo final

## Agenda

- La materia
- **Agentes**
  - **Definición**
  - Agentes inteligentes
  - Arquitecturas abstractas y concretas
  - Agentes personales – móviles
- **Sistemas multiagentes**
  - Conceptos
  - Comunicación



## Agentes

- Agentes en la vida real



## Agentes: definiciones

- Rusell & Norving

*“Un agente es cualquier cosa que pueda ver en su **entorno** a través de **sensores** y actuar en su entorno a través de **efectores**”.*

- Nwana

*“Agente: componente de software y/o hardware el cual es capaz de actuar de manera precisa con el fin de realizar tareas en **representación** de su usuario”.*



## Agentes: definiciones...

- Shoham

*“Un agente es una entidad cuyo estado es visto como compuesto de **componentes mentales** tales como creencias, elecciones, aptitudes y compromisos”.*

- Maes

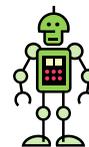
*“Un agente **autónomo** es un sistema computacional que habita en un entorno dinámico-complejo, **percibiendo** y **actuando** autónomamente en este **entorno**, y haciendo esto para realizar un conjunto de **objetivos** o tareas para los cuales fueron diseñados”.*



## Agentes: más definiciones...

- Wooldridge & Jennings

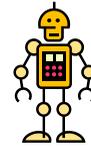
*“Un agente es un **sistema computacional** que está situado en algún **ambiente**, y que es capaz de actuar **autónomamente** en dicho ambiente con el fin de cumplir sus **objetivos**”.*



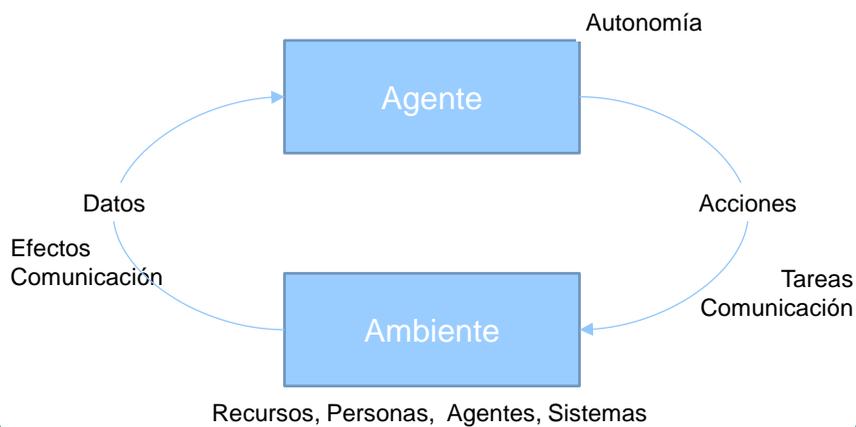
## Agentes: más definiciones...

- FIPA (Foundation for Intelligent Physical Agents)

*“Un agente es una entidad de software encapsulado con su propio **estado**, **conducta**, **hilo de control** y la **habilidad** para interactuar y **comunicarse** con otras entidades (gente, otros agentes o sistemas)”.*



## Vista abstracta de un agente



## Ejemplos de agentes

- Sistemas de control

- Termostato



- Alarma



- Software daemons

## Agentes inteligentes

- Wooldridge & Jennings

*“Un **agente inteligente** es un agente capaz de ejecutar **flexiblemente acciones autónomas** con el fin de cumplir sus **objetivos**”, donde flexibilidad significa:*

- **Reactividad:** capacidad de percibir su ambiente, y responder sin demoras a cambios que ocurren en él.
- **Pro-actividad:** capacidad de exhibir un comportamiento dirigido a objetivos, tomando la iniciativa.
- **Habilidad social:** capacidad de interactuar con otros agentes (y posiblemente humanos) a través de un lenguaje de comunicación.

## Agentes inteligentes

- |                    |              |
|--------------------|--------------|
| ● Características  | ● Termostato |
| ● Autonomía        | Si           |
| ● Reactividad      | Si           |
| ● Pro-actividad    | No           |
| ● Habilidad social | No           |



## Otras cuestiones...

- Adaptación
  - Implica percibir el entorno y reconfigurar el estado mental en respuesta a este entorno.
- Aprendizaje
  - Mediante distintas técnicas de aprendizaje de máquina.

## Agentes y objetos

- **Objetos:**  
Entidades computacionales que *encapsulan* algún estado, son capaces de realizar acciones, o *métodos* en este estado, y se comunican mediante el *envío de mensajes*.
- **Agentes vs. Objetos**
  - Autonomía
    - POO: decisión en el objeto que invoca el método
    - MAS: decisión en el agente que recibe el pedido
  - Conducta flexible
  - Control

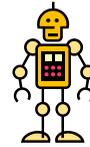


## Agentes y sistemas expertos

- **Sistema experto**  
Sistema capaz de resolver problemas o dar consejos en algún dominio rico en conocimiento.
- **Agentes vs. SE**
  - SE no interactúan directamente con el ambiente.

## Programación Orientada a Agentes

- Paradigma de programación centrado en el concepto de agentes de software.

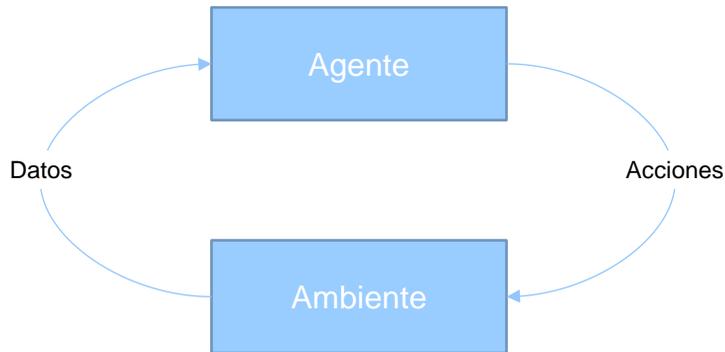


## Agenda

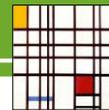
- La materia
- Agentes
  - Definición
  - Agentes inteligentes
  - **Arquitecturas abstractas y concretas**
  - Agentes personales – móviles
- Sistemas multiagentes
  - Conceptos
  - Comunicación



## Vista abstracta de un agente

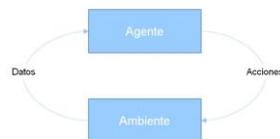


## Arquitectura abstracta de un agente inteligente



- Ambiente  
 $S = \{s_1, s_2, \dots\}$
- Acciones  
 $A = \{a_1, a_2, \dots\}$
- Historia
- Agente estándar  
 $action: S^* \rightarrow A$
- Ambiente no determinista  
 $env: S \times A \rightarrow \wp(S)$

$$h: s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots \xrightarrow{a_{u-1}} s_u \xrightarrow{a_u} \dots$$



## Agentes puramente reactivos

- Toma de decisiones enteramente basada en el presente.

*action*:  $S \rightarrow A$

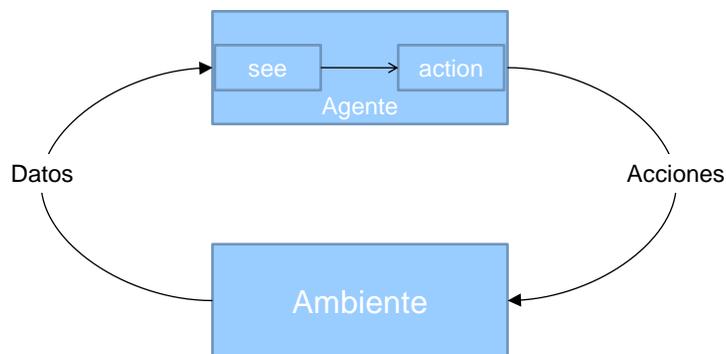
- Termostato

- $S = \{s \geq temperatura, s < temperatura\}$

- $A = \{calefacción\_on, calefacción\_off\}$

$$action(s) = \begin{cases} calefacción\ off & \text{si } s \geq temperatura \\ calefacción\ on & \text{cualquier otra} \end{cases}$$

## Percepción y acción



$see : S \rightarrow P$

$action : P^* \rightarrow A$

## Percepción y acción

### ● Ventajas de la división

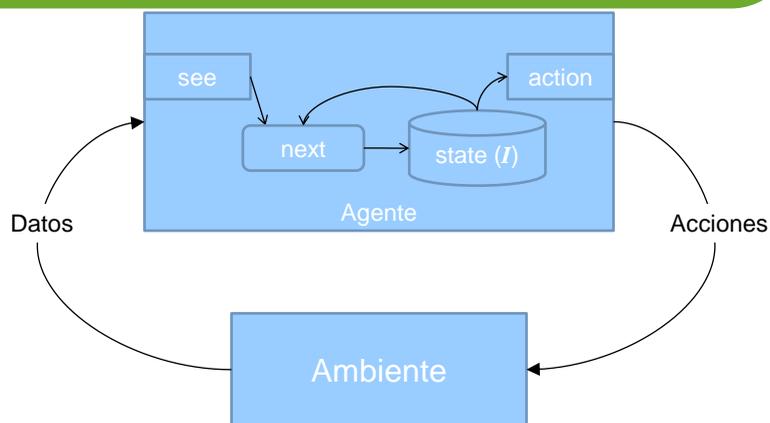
#### ● Sea

- X: “La temperatura de la habitación es Ok”.
- Y: “Ventana cerrada”.

$$S = \{\{-x, \neg y\}, \{-x, y\}, \{x, \neg y\}, \{x, y\}\}$$

$$see(s) = \begin{cases} p_1 & \text{si } s = s_1 \\ p_2 & \text{si } s = s_2 \\ p_3 & \text{si } s = s_3 \text{ o } s_4 \end{cases} \quad action(p) = \begin{cases} \text{cerrar vent} & p_1 \\ \text{calefacción on} & p_2 \\ \text{calefacción off} & p_3 \end{cases}$$

## Agentes con estado



$$see : S \rightarrow P \quad action : I \rightarrow A \quad next : I \times P \rightarrow I$$

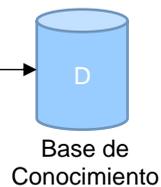
## Arquitecturas concretas

- Agentes basados en lógica
- Agentes reactivos
- Agentes BDI



## Agentes basados en lógica

- Jess
- Prolog



$\rho$  : reglas de deducción

$\Delta \vdash_p \phi$

$see : S \rightarrow P$      $next : D \times P \rightarrow D$      $action : D \rightarrow A$

## Agentes basados en lógica

```

1. function action( $\Delta:D$ ):A
2. begin
3.   for each  $a \in A$  do
4.     if  $\Delta \vdash_p Do(a)$  then
5.       return  $a$ 
6.     endif
7.   endfor
8.   for each  $a \in A$  do
9.     if  $\Delta \not\vdash_p \neg Do(a)$  then
10.      return  $a$ 
11.    endif
12.  endfor
13. return null
14. end function action

```

## Arquitectura reactiva

- Toma de decisiones
  - Conjunto de comportamientos para llevar a cabo tareas.
  - Comportamiento
    - Situación  $\rightarrow$  Acción
- Comportamientos simultáneos
  - Jerarquía de comportamientos
    - Prioridad

## Arquitectura reactiva

- Definición formal de un comportamiento

$$Beh = \{(c, a) \mid c \subseteq P \wedge a \in A\}$$

- Un comportamiento  $(c, a)$  será disparado cuando el ambiente esté en el estado  $s$  perteneciente a  $S$  si  $see(s)$  pertenece a  $c$

- Relación de inhibición

$$R \subseteq Beh \qquad \prec \subseteq R \times R$$

## Arquitectura reactiva

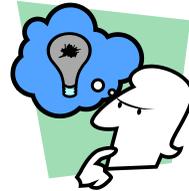
```

1. function action( $p:P$ ): $A$ 
2.   var fired:  $\wp(R)$ 
3.
4.   begin
5.     fired :=  $\{(c, a) \mid (c, a) \in R \wedge p \in c\}$ 
6.     for each  $(c, a) \in \textit{fired}$  do
7.       if  $\neg(\exists(c', a') \in \textit{fired}$  tal que  $(c', a') \prec (c, a))$  then
8.         return  $a$ 
9.       endif
10.    endfor
11.    return null
12.  end function action

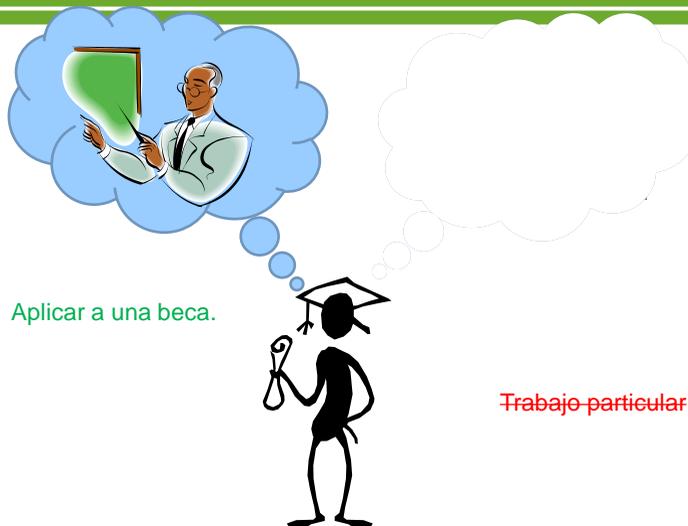
```

## Arquitectura BDI

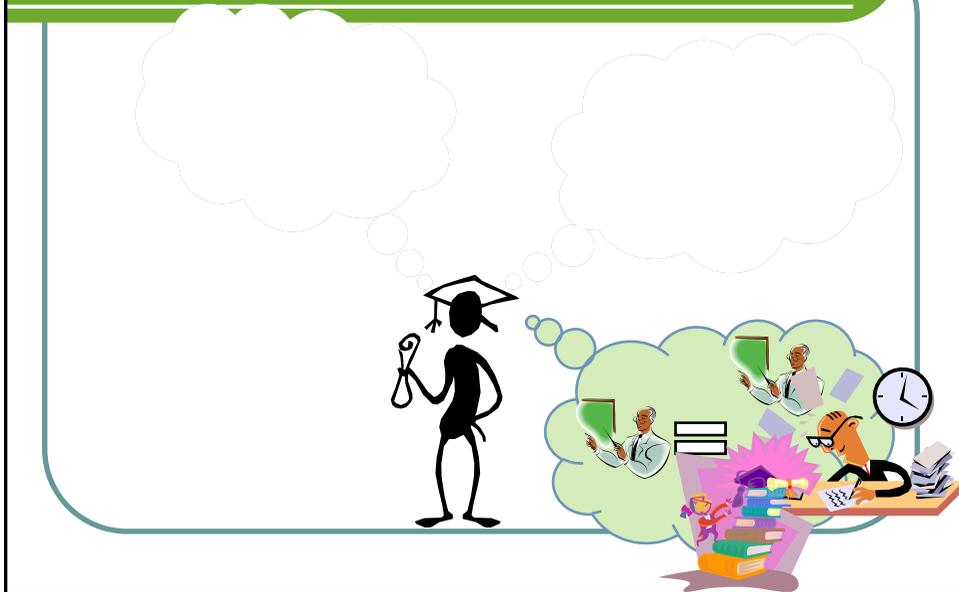
- **Belief-Desire-Intention**
- Razonamiento práctico
  - Deliberación
    - ¿Qué objetivos cumplir?
  - Razonamiento orientado a objetivos
    - ¿Cómo se cumplirán los objetivos?



## Ejemplo de razonamiento práctico



## Ejemplo de razonamiento práctico



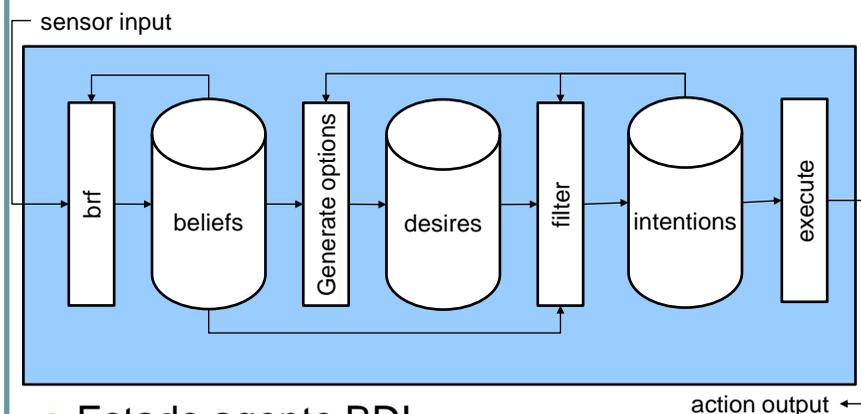
## Intenciones

- Rol de las intenciones en el razonamiento práctico
  - Dirigen el razonamiento basado en medios y fines (orientado a objetivos)
  - Restringen la deliberación futura
  - Persisten
  - Influencian las creencias bajo las cuales se basa el razonamiento práctico futuro

## Intenciones

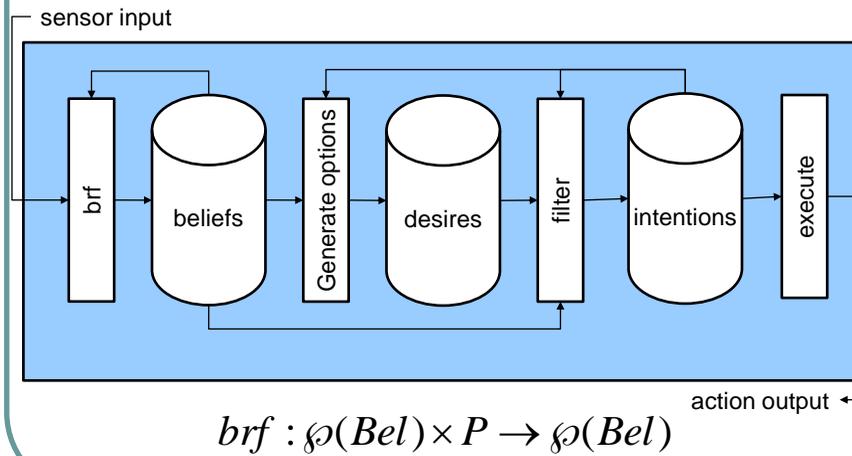
- Reconsideración de intenciones
  - Un agente que no se detiene a reconsiderar sus intenciones (reactivo – orientado a eventos).
  - Un agente que constantemente reconsidera sus intenciones (proactivo – orientado a objetivos).
  
- Taza de cambio en el mundo  $\gamma$ 
  - Si  $\gamma$  es baja, los agentes que nunca se detienen a reconsiderar funcionan mejor.
  - Si  $\gamma$  es alta, los agentes que se detienen constantemente funcionan mejor.

## Proceso de razonamiento práctico en agentes BDI

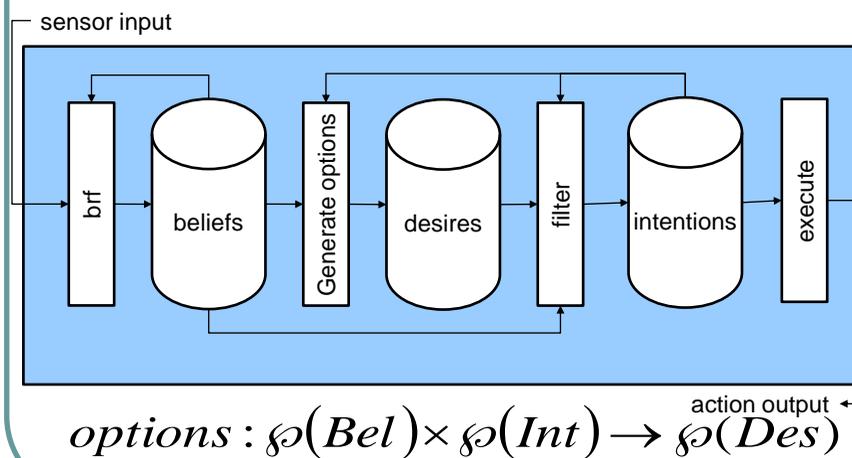


- Estado agente BDI
  - $(B, D, I)$ , donde  $B \subseteq Bel$ ,  $D \subseteq Des$ ,  $I \subseteq Int$

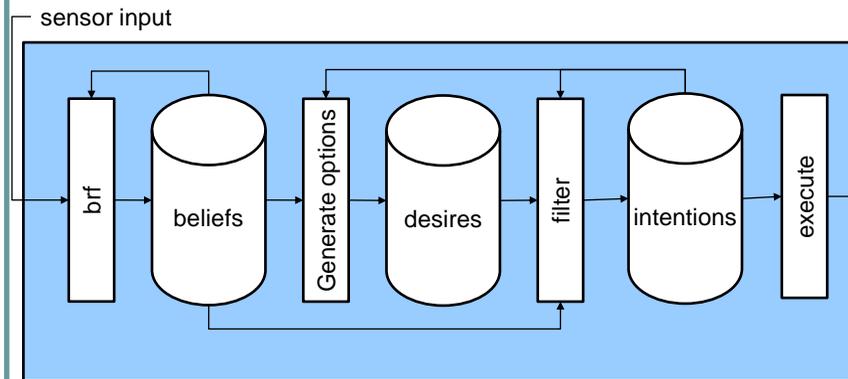
## Proceso de razonamiento práctico en agentes BDI



## Proceso de razonamiento práctico en agentes BDI



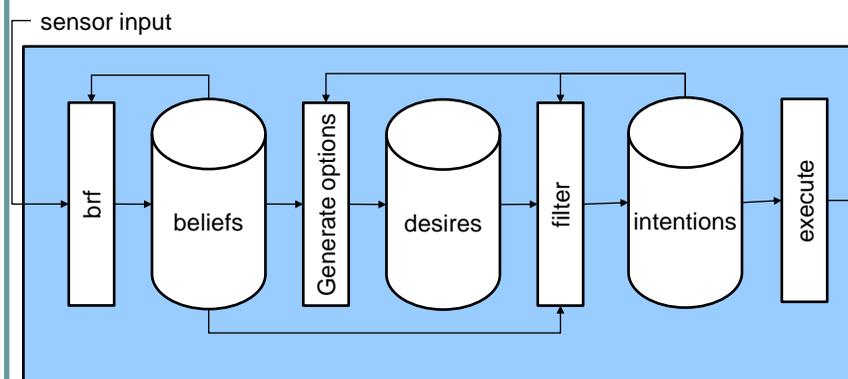
## Proceso de razonamiento práctico en agentes BDI



$$filter : \wp(Bel) \times \wp(Des) \times \wp(Int) \rightarrow \wp(Int)$$

$$\forall B \in \wp(Bel), \forall D \in \wp(Des), \forall I \in \wp(Int), filter(B, D, I) \subseteq I \cup D$$

## Proceso de razonamiento práctico en agentes BDI



$$execute : \wp(Int) \rightarrow A$$

$$action : P \rightarrow A$$

## Función BDI *action*

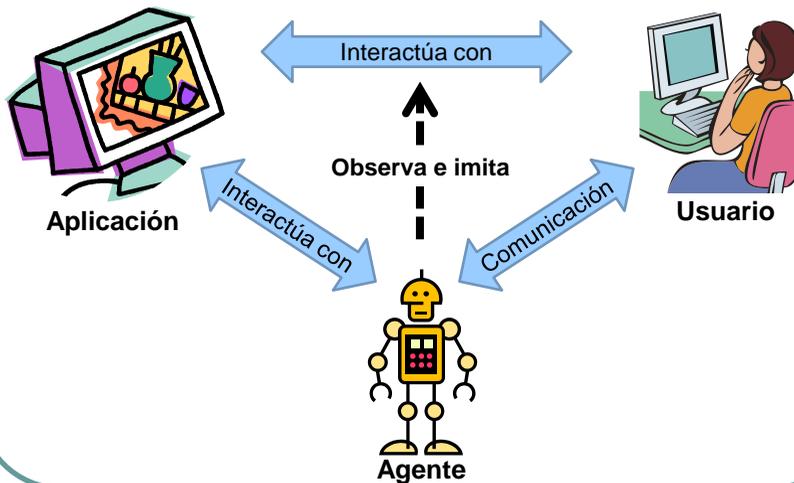
```
1. function action( $p:P$ ): $A$ 
2. begin
3.      $B := brf(B, p)$ 
4.      $D := options(B, I)$ 
5.      $I := filter(B, D, I)$ 
6.     return execute( $I$ )
7. end function action
```

## Agenda

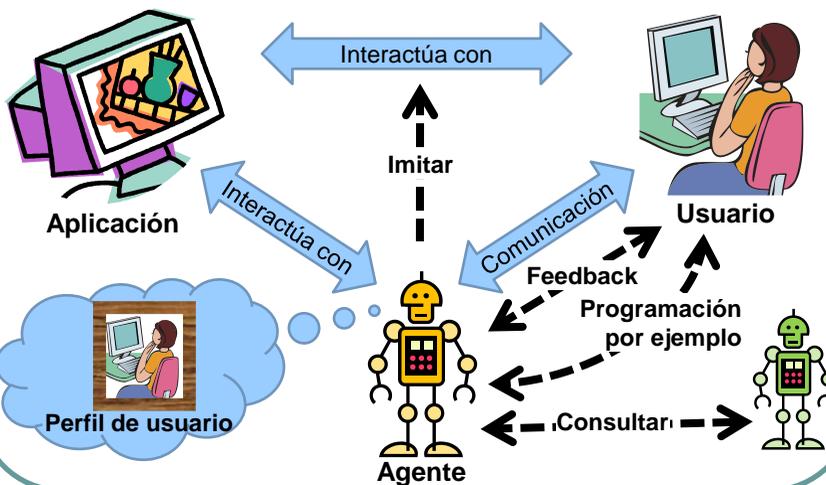
- La materia
- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitecturas abstractas y concretas
  - **Agentes personales – móviles**
- Sistemas multiagentes
  - Conceptos
  - Comunicación



## Agentes personales

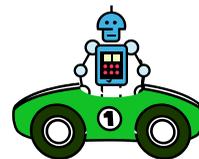


## Agentes personales: aprendizaje



## Agentes móviles

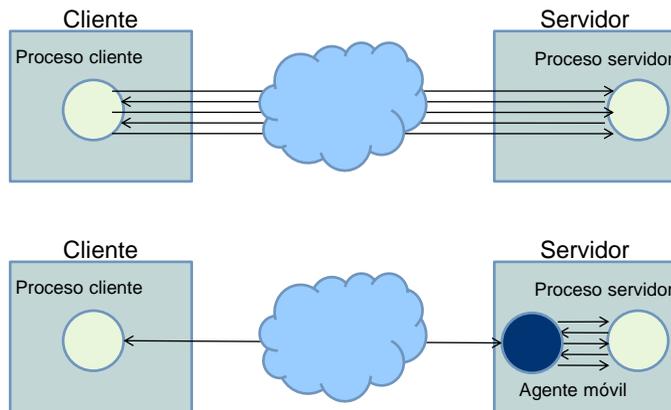
- Son agentes capaces de transmitirse a si mismos (código y estado) a través de una red de computadoras y volver a empezar su ejecución en un sitio remoto.



## Agentes móviles

- Movilidad de código:
  - capacidad de cambiar dinámicamente las ligaduras entre fragmentos de código y la ubicación donde son ejecutados.
  - Movilidad Fuerte:
    - los threads en ejecución migran de una plataforma a otra y continúan su ejecución en forma transparente.
  - Movilidad Débil
    - El código migra pero no su estado de ejecución.

## RPC vs. Agentes móviles



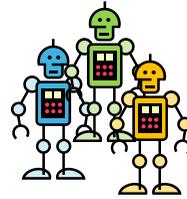
## Agenda

- La materia
- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitecturas abstractas y concretas
  - Agentes personales – móviles
- **Sistemas multiagentes**
  - Conceptos
  - Comunicación



## Sistemas multiagentes

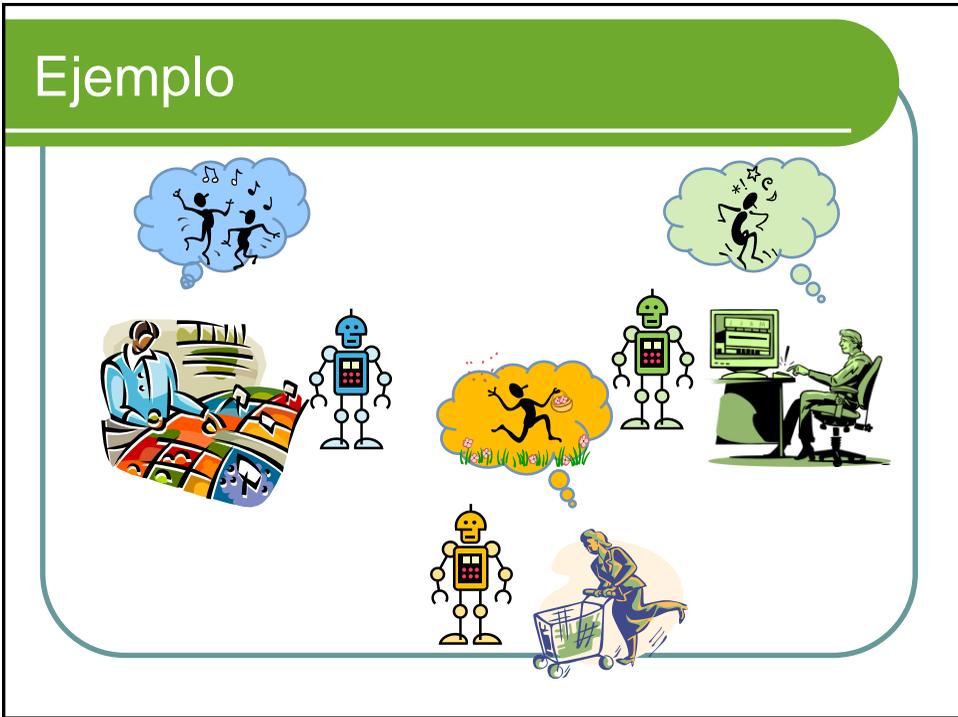
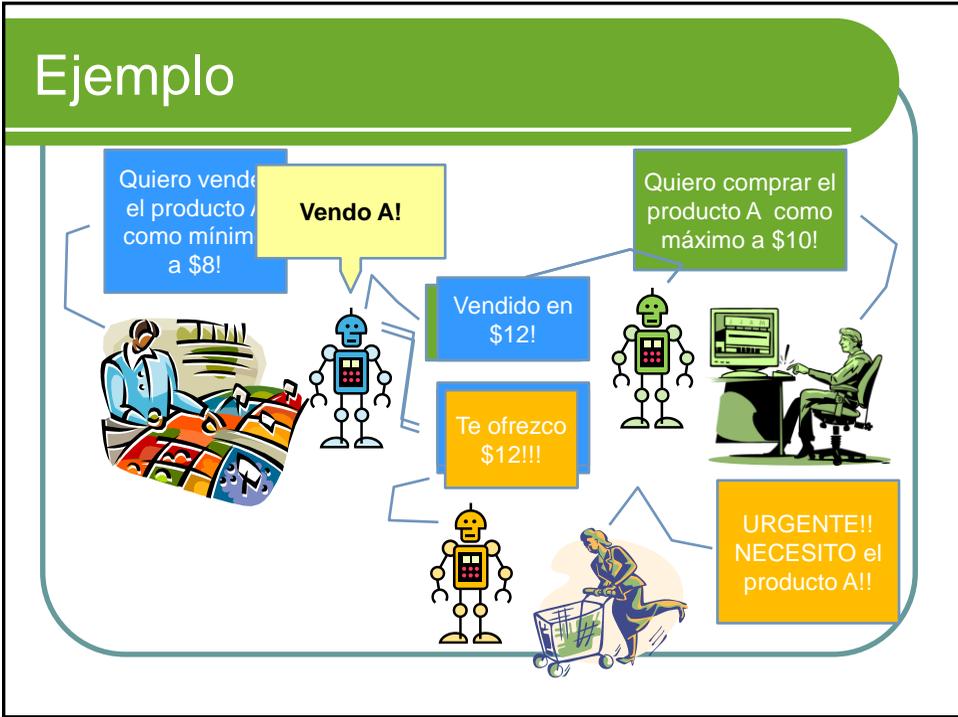
- Un sistema multiagente es un sistema compuesto de múltiples agentes inteligentes, capaces de cumplir objetivos que son difíciles de alcanzar por un sistema individual.



## Ejemplo

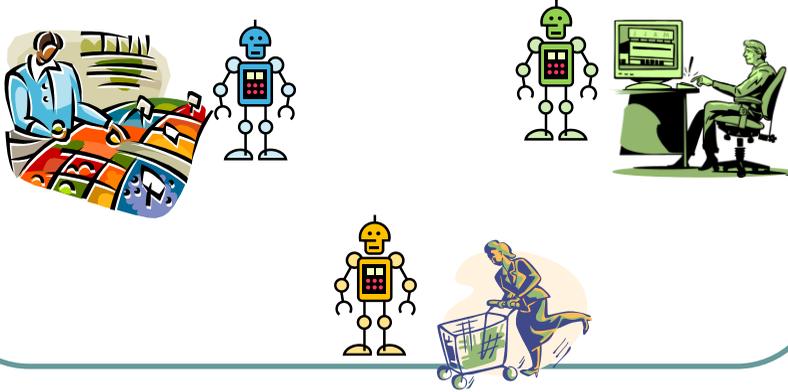


- No controladores
- No organizadores
- Auto-organización
- Reglas simples
  - Mantener distancia
  - Ser sociables
- Comportamiento complejo



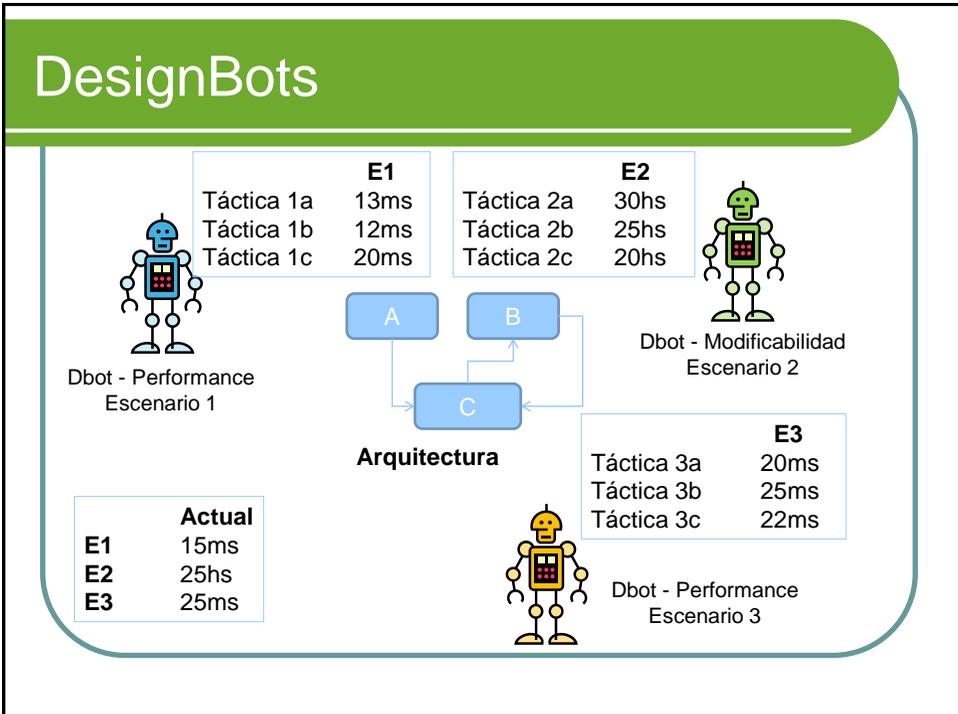
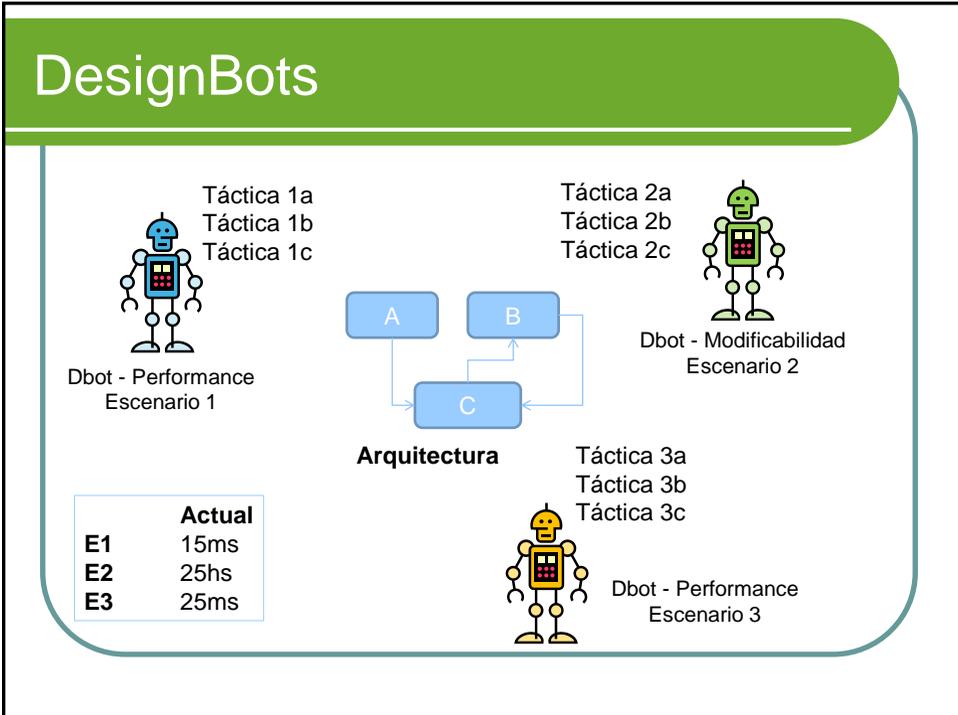
## Ejemplo

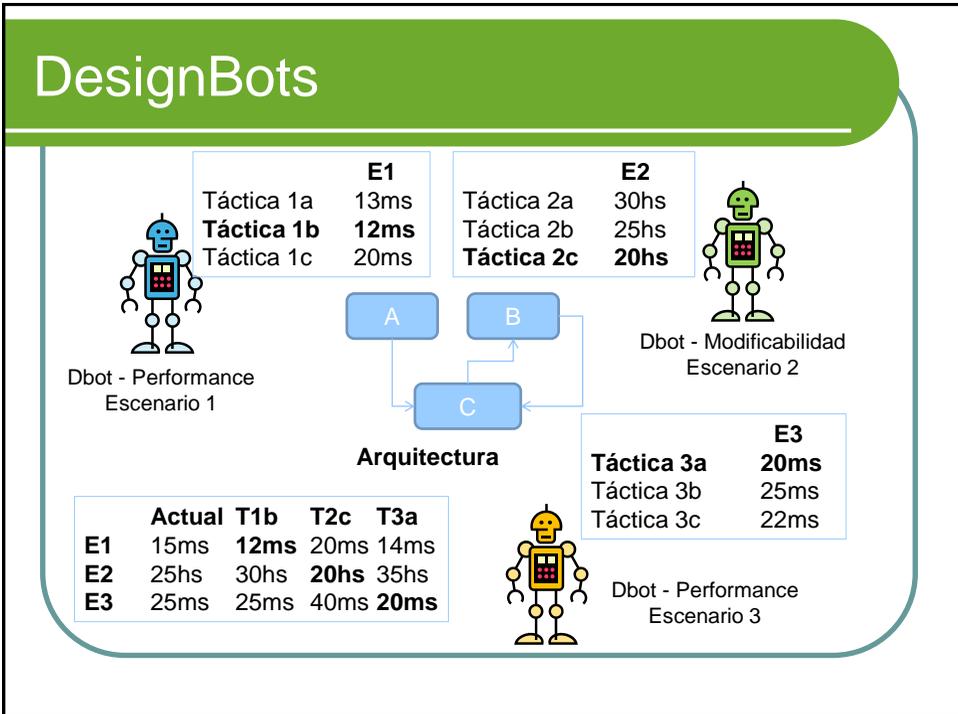
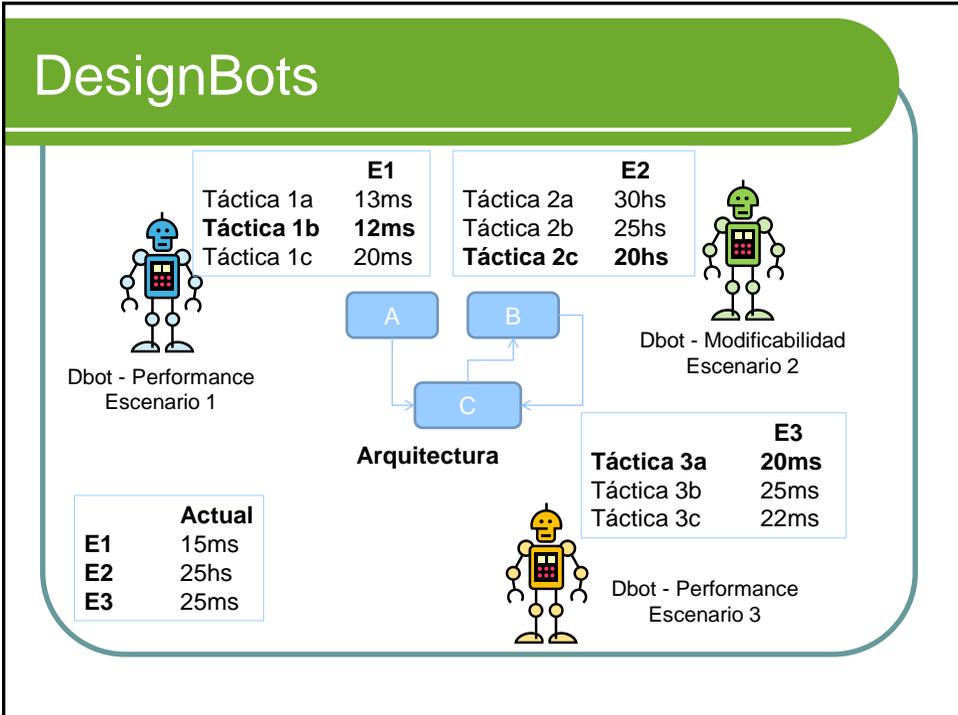
- Información privada
- Competitividad
- Confianza
- Reputación

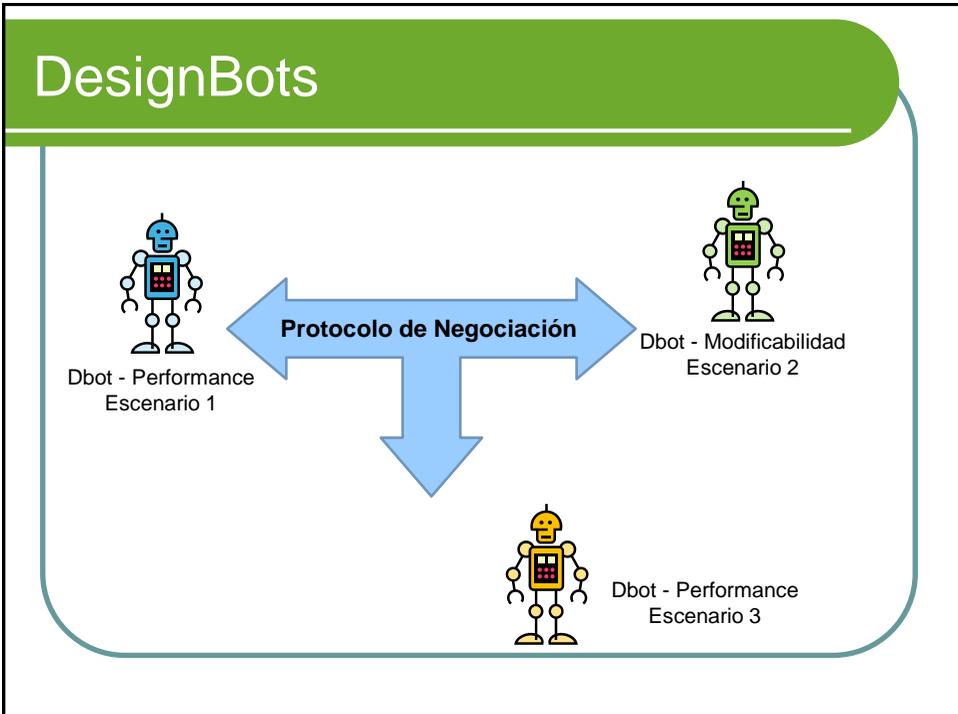
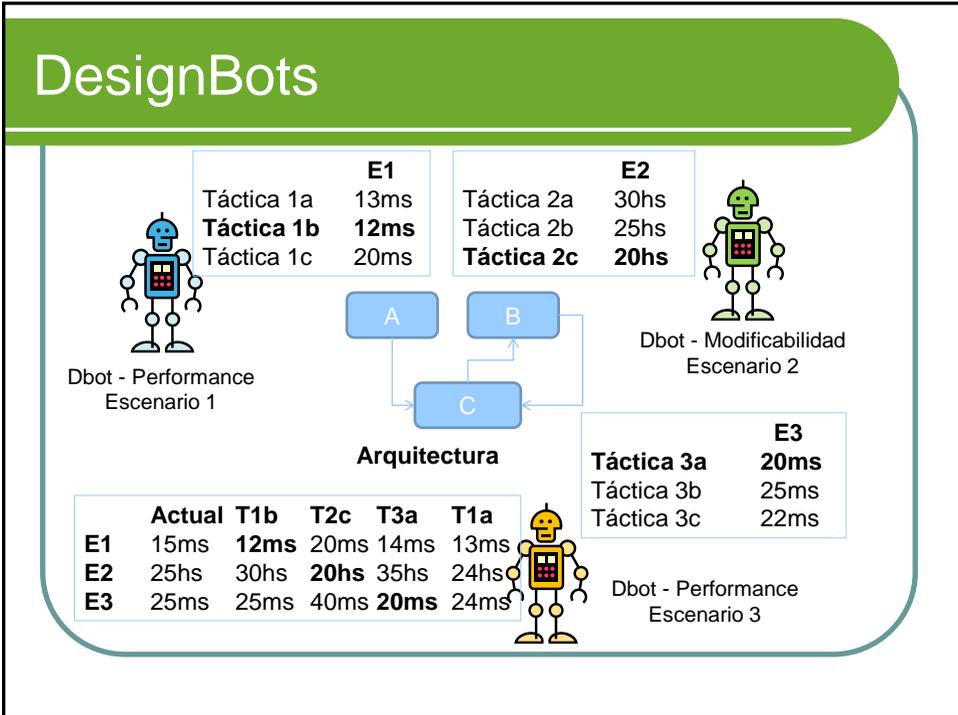


## Ejemplo cooperativo





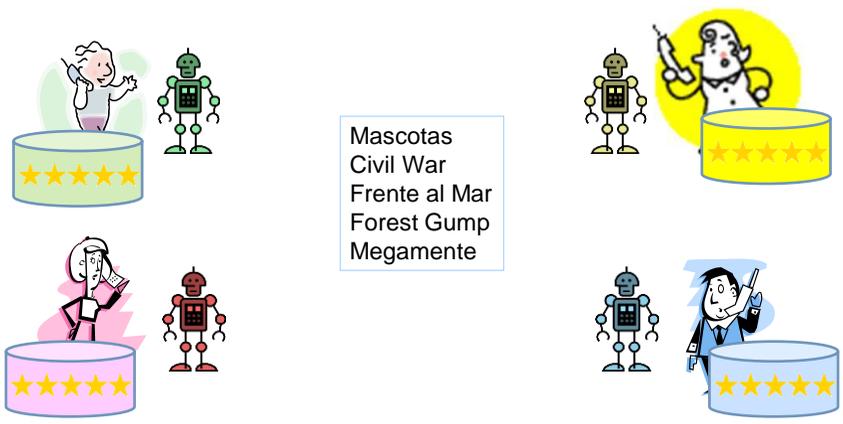




# PUMAS-GR



# PUMAS-GR



# PUMAS-GR



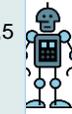
Mascotas 5  
Megamente 4,5  
Civil War 4  
Forest Gump 2  
Frente al Mar 1

Forest Gump 5  
Frente al Mar 4  
Megamente 4  
Mascotas 2  
Civil War 1



Frente al Mar 5  
Forest Gump 4  
Megamente 4  
Mascotas 3  
Civil War 1

Civil War 5  
Megamente 4,5  
Mascotas 3  
Frente al Mar 2  
Forest Gump 2



# PUMAS-GR

**Mascotas 5**  
Megamente 4,5  
Civil War 4  
Forest Gump 2  
Frente al Mar 1

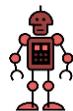


Mascotas

Forest Gump



**Forest Gump 5**  
Frente al Mar 4  
Megamente 4  
Mascotas 2  
Civil War 1



Frente al Mar

Civil War



**Civil War 5**  
Megamente 4,5  
Mascotas 3  
Frente al Mar 2  
Forest Gump 2

¿Quién empieza?

# PUMAS-GR

**Mascotas 5**  
Megamente 4,5  
Civil War 4  
Forest Gump 2  
Frente al Mar 1

**Acepto si:  $u(mi\_p) \leq u(otra\_p)$**

$u(mascotas) \leq u(forestgump)?$   
 $5 \leq 2?$   
 $u(mascotas) \leq u(frentemar)?$   
 $5 \leq 1?$   
 $u(mascotas) \leq u(civilwar)?$   
 $5 \leq 4?$

**Forest Gump 5**  
Frente al Mar 4  
Megamente 4  
Mascotas 2  
Civil War 1

**Frente al Mar 5**  
Forest Gump 4  
Megamente 4  
Mascotas 3  
Civil War 1

**¿Cómo evaluó?**

**Civil War 5**  
Megamente 4,5  
Mascotas 3  
Frente al Mar 2  
Forest Gump 2

# PUMAS-GR

**Mascotas 5**  
Megamente 4,5  
Civil War 4  
Forest Gump 2  
Frente al Mar 1

NO!

NO!

**Forest Gump 5**  
Frente al Mar 4  
Megamente 4  
Mascotas 2  
Civil War 1

**Frente al Mar 5**  
Forest Gump 4  
Megamente 4  
Mascotas 3  
Civil War 1

NO!

NO!

**Civil War 5**  
Megamente 4,5  
Mascotas 3  
Frente al Mar 2  
Forest Gump 2

# PUMAS-GR

**Mascotas 5**  
Megamente 4,5  
Civil War 4  
Forest Gump 2  
Frente al Mar 1

**¿Alguien debe conceder?  
¿Quién?**

$Z=(5-1)/5=0,8$

$Z=(5-1)/5=0,8$

$$Z_i = \begin{cases} 1 & \text{si } u_i(x_i) = 0 \\ \frac{u_i(x_i) - \min\{u_i(x_k) | k \in A\}}{u_i(x_i)} \end{cases}$$

$Z=(5-1)/5=0,8$

$Z=(5-2)/5=0,6$

**Forest Gump 5**  
Frente al Mar 4  
Megamente 4  
Mascotas 2  
Civil War 1

**El agente con menor valor Zeuthen concede.**

**Frente al Mar 5**  
Forest Gump 4  
Megamente 4  
Mascotas 3  
Civil War 1

**Civil War 5**  
Megamente 4,5  
Mascotas 3  
Frente al Mar 2  
Forest Gump 2

# PUMAS-GR

**Mascotas 5**  
Megamente 4,5  
Civil War 4  
Forest Gump 2  
Frente al Mar 1

**¿Cómo conceder?**

$u(\text{megamente}) \leq u(\text{megamente})?$   
 $4.5 \leq 4.5?$

$u(\text{frentemar}) \leq u(\text{megamente})?$   
 $4 \leq 4?$

Megamente?

$u(\text{forestgump}) \leq u(\text{megamente})?$   
 $4 \leq 4?$

**Forest Gump 5**  
Frente al Mar 4  
Megamente 4  
Mascotas 2  
Civil War 1

**Evaluación alternativa:  
Acepto si:  $u(\text{prox}_p) \leq u(\text{otra}_p)$**

**Frente al Mar 5**  
Forest Gump 4  
Megamente 4  
Mascotas 3  
Civil War 1

**Civil War 5**  
~~Megamente 4,5~~  
Mascotas 3  
Frente al Mar 2  
Forest Gump 2

## Características de ambientes multiagentes

- Proveen una infraestructura específica de comunicación y protocolos de interacción.
- Son típicamente abiertos y no tienen un diseño centralizado.
- Contienen agentes que son autónomos y distribuidos, y pueden ser interesados en si mismos o cooperativos.

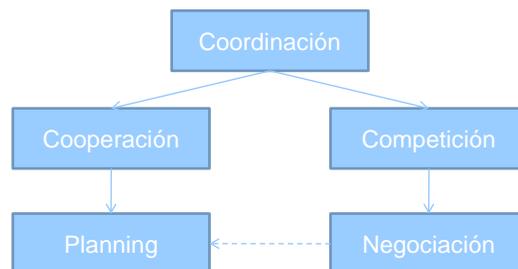
## Agenda

- La materia
- Agentes
  - Definición
  - Agentes inteligentes
  - Arquitecturas abstractas y concretas
  - Agentes personales – móviles
- Sistemas multiagentes
  - Conceptos
  - **Comunicación**



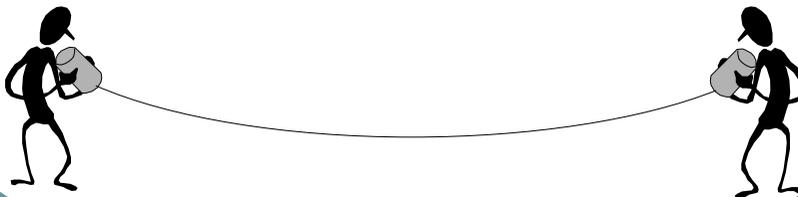
## Comunicación entre agentes

- Percepción
  - Recibir mensajes
- Acción
  - Enviar mensajes



## Agent Communication Languages (ACL)

- KIF
- KQML
- FIPA ACL
- Ontologías



## KSE: Knowledge Sharing Effort

- Iniciado por DARPA hacia 1990, y apoyado por organismos norteamericanos de investigación (ASOFR, NSF, NRI)
  - Propósito:
    - Desarrollo de técnicas, metodologías y herramientas software para compartir y utilizar el conocimiento entre sistemas a lo largo de las etapas del ciclo de vida del software.
  - KIF y KQML

## KIF

- Originalmente desarrollado con la intención de ser un lenguaje común para expresar propiedades de un dominio en particular.
  - No mensajes – Si contenido
  - Ejemplo
    - (= (temperature ml) (scalar 83 Celsius))
    - (defrelation bachelor (?x) :=  
 (and (man ?x)  
 (not (married ?x))))



## KQML

- KQML es un lenguaje y protocolo basado en mensajes para comunicación entre agentes.

- Cada mensaje posee

- Performativa
- Parámetros

```
(ask-one
  :content (PRICE IBM ?price)
  :receiver -stock-server
  :language LPROLOG
  :ontology NYSE-TICKS)
```

## Mensajes KQML

- :content
  - Contenido del mensaje
- :force
  - Si el agente quiere denegar el contenido de un mensaje
- :reply-with
  - Identificador de la respuesta esperada por el agente
- :in-reply-to
  - Referencia a :reply-with
- :sender
  - Emisor del mensaje
- :receiver
  - Receptor del mensaje

## FIPA ACL

- Lenguaje para intercambio de mensajes entre agentes.
- 20 performativas
  - accept-proposal, agree, cancel, cfp, confirm, inform, refuse, request, etc.

(inform

```
:sender agent1
:receiver agent2
:content (price good2 100)
:language sl
:ontology hpl-auction)
```

## Ontologías para la comunicación entre agentes

- Necesidad de acordar terminología.
  - “El agente cometió un error.”



## Ontologías para la comunicación entre agentes

- Necesidad de acordar terminología.
  - “El agente cometió un error.”



## Ontologías para la comunicación entre agentes

- Necesidad de acordar terminología.
  - “El agente cometió un **error**.”



## Web Semántica vs. Web Actual

- Web actual representa la información utilizando documentos en lenguaje natural con poca estructura
  - Fácil comprensión por humanos (*html* sólo define presentación).
  - Difícil soportar el procesamiento automático.
- Alternativas
  - Máquinas más inteligentes → comprender el significado de la información que hay en la web
    - Procesamiento de lenguaje natural, reconocimiento de imágenes, etc.
  - Información más inteligente → representar la información de modo que sea sencilla de comprender a las máquinas
    - Expresar contenidos en un formato procesable automáticamente.
    - Uso de **metainformación (metadatos = datos sobre los datos)**

## Definición

« Una **especificación de una conceptualización**... una descripción de los conceptos y relaciones que pueden existir para un agente o una comunidad de agentes. »

*T.R. Gruber*

« Una ontología es un catalogo de los tipos de cosas que, se asume, existen en un dominio de interés D desde la perspectiva de una persona, la cual usa un lenguaje L para hablar sobre D. »

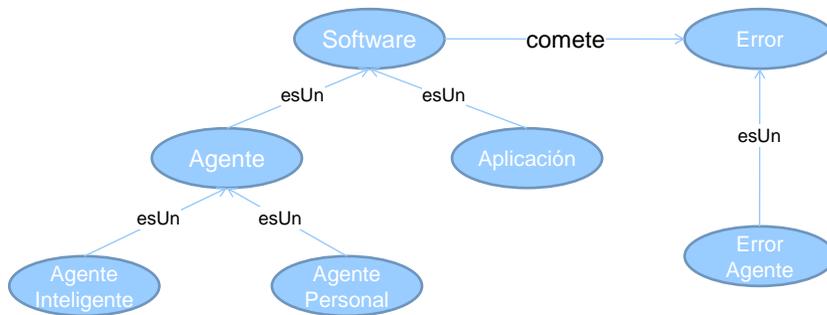
*John F. Sowa*

« *Un ontología trata sobre la exacta descripción de las cosas y sus relaciones.* »

*World Wide Web Consortium (W3C)*

## Ejemplo de ontología

- “El agente cometió un error.”



## Resumen

- Agentes inteligentes
  - Autonomía – Reactividad – Proactividad – Habilidad social.
- Ambientes multiagentes
  - Comunicación
  - Agentes heterogéneos
  - Cooperación – Competición

## Bibliografía

- Jennings, N.R., On agent-based software engineering. *Artificial Intelligence*, 117, 2000.
- P. Maes, 1994. *Agents that reduce work and information overload*. *Communications of the ACM*, vol.37(7), pp.30-40, 1994.
- Weiss, G. *Multiagent Systems*. The MIT Press, 1999.
- M. Wooldridge *An Introduction to MultiAgent Systems - Second Edition*, Wiley & Sons, 2009.