

Agentes

Jose Aguilar

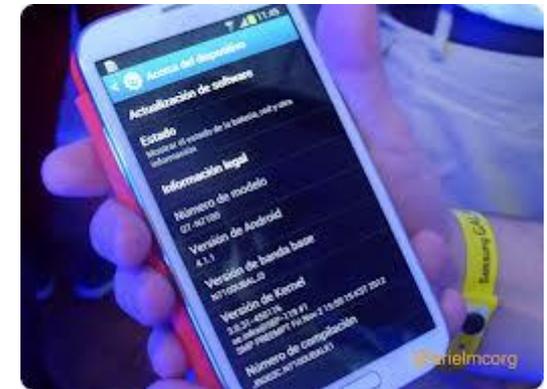
Cemisid, Facultad de Ingeniería

Universidad de los Andes

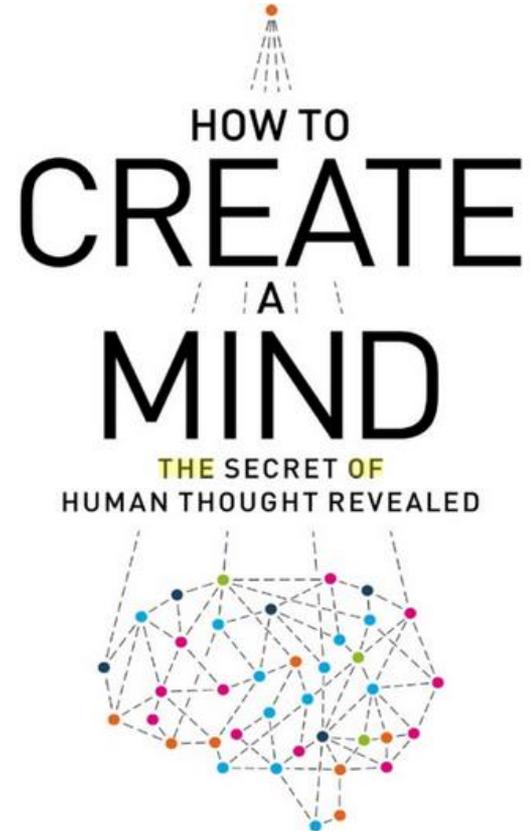
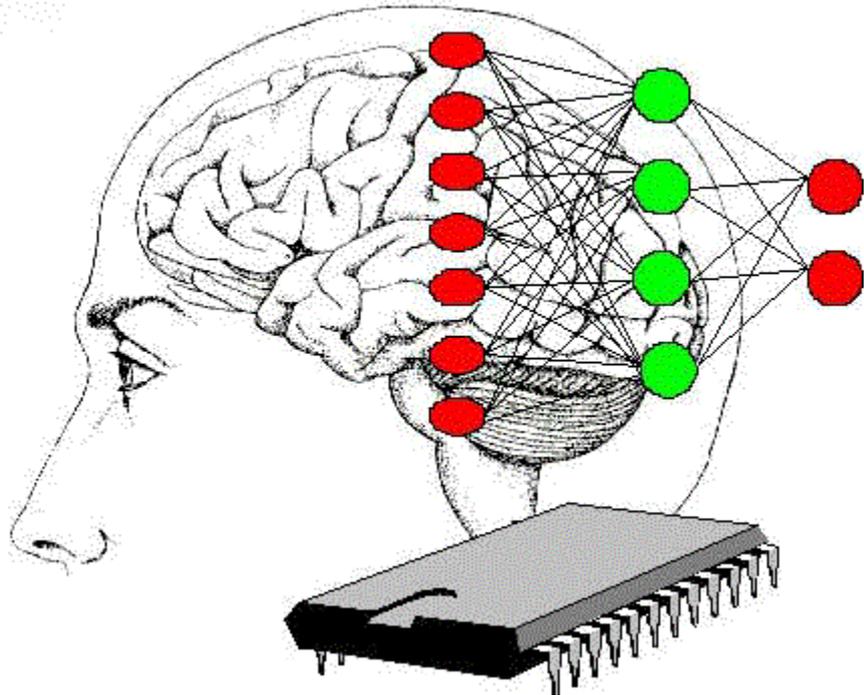
Mérida, Venezuela

aguilar@ula.ve

Aplicaciones



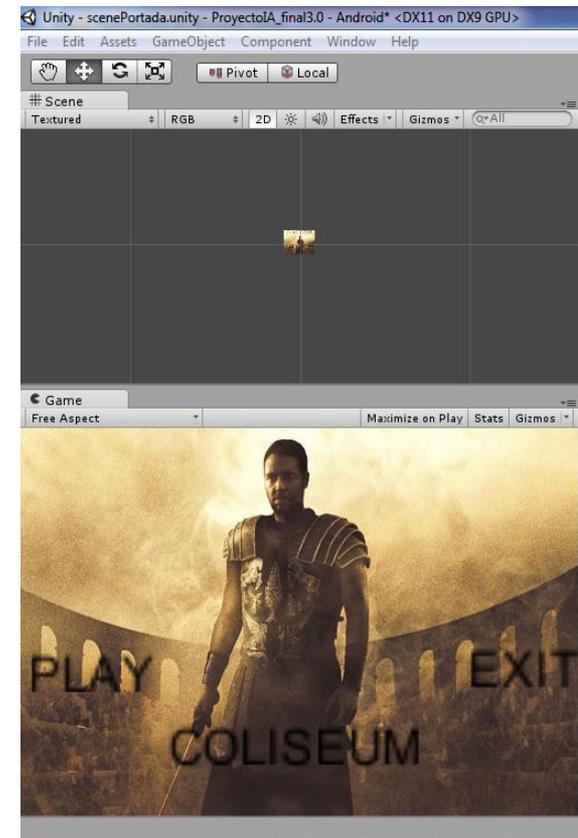
Retos



Agente gladiador

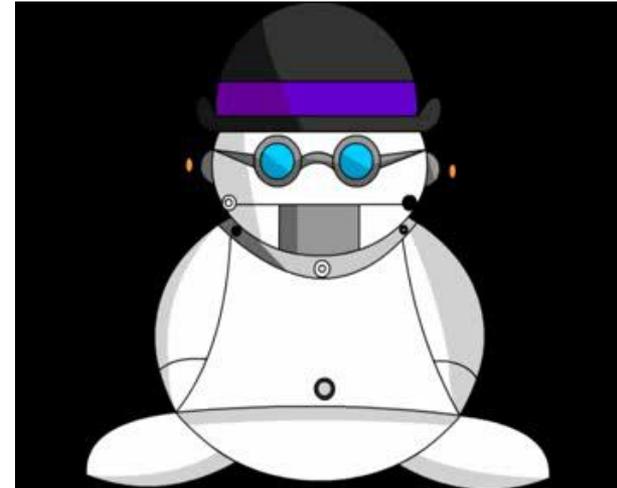
Este es un agente que juega COLISEUM de forma inteligente este juego consiste en un duelo entre dos personajes el agente y un jugador los cuales lucharán a muerte en la arena usando espadas.

Durante el juego aparecerá una botiquín de forma aleatoria, conseguir la curación que este proporciona es objetivo importante de el juego



Agente Go

Este es un agente que debe ser capaz de ayudar, avisar y/o aconsejar sobre las actividades pendientes de acuerdo a las materias seleccionada durante el semestre por un estudiante



Agente

Definición de la real academia de la lengua española
(www.rae.es):

Persona o cosa que produce un efecto y que obra con poder de otra.

Tres elementos fundamentales;

1. **Produce** un efecto,
2. **Obra** (ejecuta alguna acción)
3. Lo hace **en función de otro**.

Definición de un Agente

Shoham: “una entidad que *funciona continua y autónomamente*, en un *entorno* en el cual otros procesos ocurren y existen otros agentes” .

Russel: “entidad que *percibe su entorno y actúa* bajo estas percepciones”.

Ferber: “hardware o software que *puede actuar* sobre si mismo o sobre su ambiente. Además, tiene una *representación parcial* de su ambiente y *puede comunicarse* con otros agentes. Por otro lado, tiene *objetivos individuales* y su *comportamiento es el resultado* de las observaciones, conocimiento, habilidades e interrelaciones que él puede tener con otros agentes o con su ambiente”.

Agentes

Los agentes de software son capaces de *decidir por sí mismos* qué hacer en una determinada situación. *Mantienen información* acerca de su entorno, y *toman decisiones* en función de *su percepción* del estado de dicho entorno, sus experiencias anteriores, y los objetivos que tienen planteados. Además, los agentes *pueden comunicarse* con otros agentes para colaborar y alcanzar objetivos comunes [Denney 2008]

Agente

Es cualquier cosa que pueda **percibir** su ambiente a través de *sensores*, y **actuar** sobre él a través de *actuadores*

- Agente Humano: ojos, oídos, manos, brazos, etc.,
- Agente Robot: cámaras, sensores y varios motores actuadores

Agente

ES TODA ENTIDAD FISICA O VIRTUAL QUE:

1. ES CAPAZ DE **INTERACTUAR** CON SU AMBIENTE
2. PUEDE **COMUNICARSE** DIRECTAMENTE CON OTROS AGENTES
3. TIENE UN CONJUNTO DE DESEOS (P.E.: BAJO LA FORMA DE **OBJETIVOS INDIVIDUALES O FUNCIONES DE SATISFACCION**, QUE BUSCA OPTIMIZAR)
4. POSEE **RECURSOS PROPIOS**
5. ES CAPAZ DE **PERCIBIR** SU AMBIENTE
6. TIENE UNA **REPRESENTACION** PARCIAL DE SU AMBIENTE
7. POSEE CIERTAS **COMPETENCIAS** Y DA **SERVICIOS**
8. PUEDE **REPRODUCIRSE**
9. SU COMPORTAMIENTO TRATA DE **SATISFACER SUS OBJETIVOS**

Agente

Es un sistema computacional que está situado en un entorno, que es capaz de realizar acciones autónomas flexibles en ese entorno para alcanzar sus objetivos

- **Caracterizado por:**

- SU ESTRUCTURA (ARQUITECTURA)
- SUS ACCIONES (COMPORTAMIENTO)

Arquitectura+programa

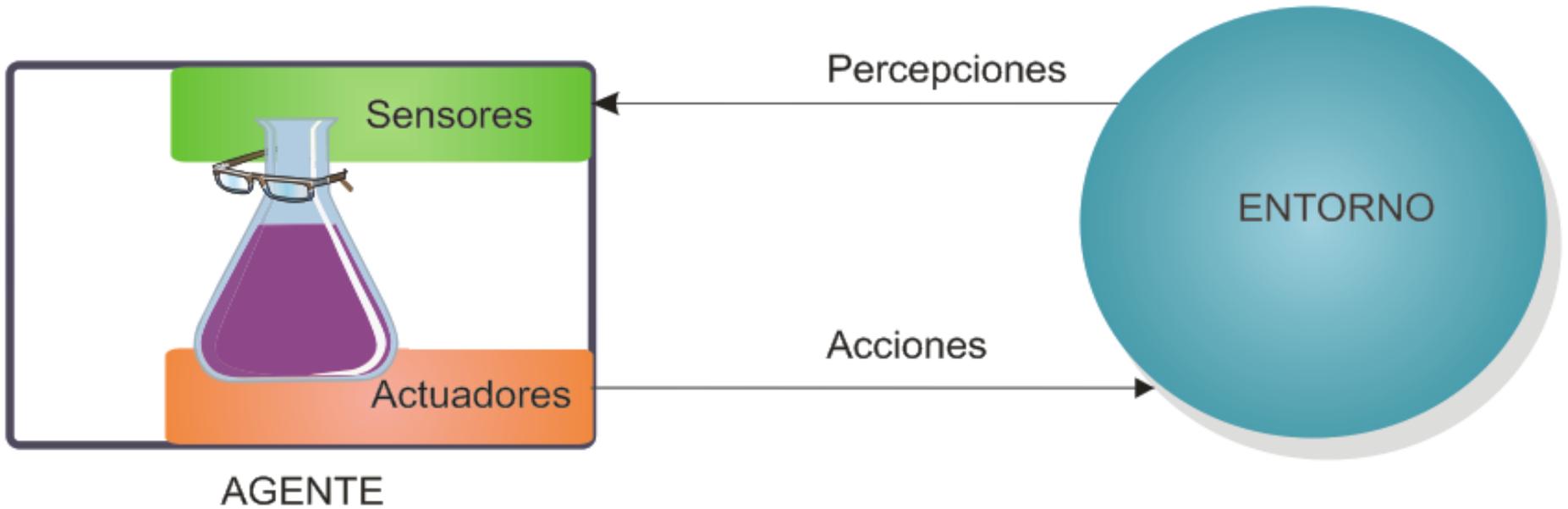
Estructura de los Agentes

Agente = programa + arquitectura

- El programa implementará la transformación (mapeo) de las secuencias de percepciones en acciones.
- La arquitectura será un computador (equipo) que se ocupará de que las percepciones lleguen al programa y las acciones lleguen a los efectores.

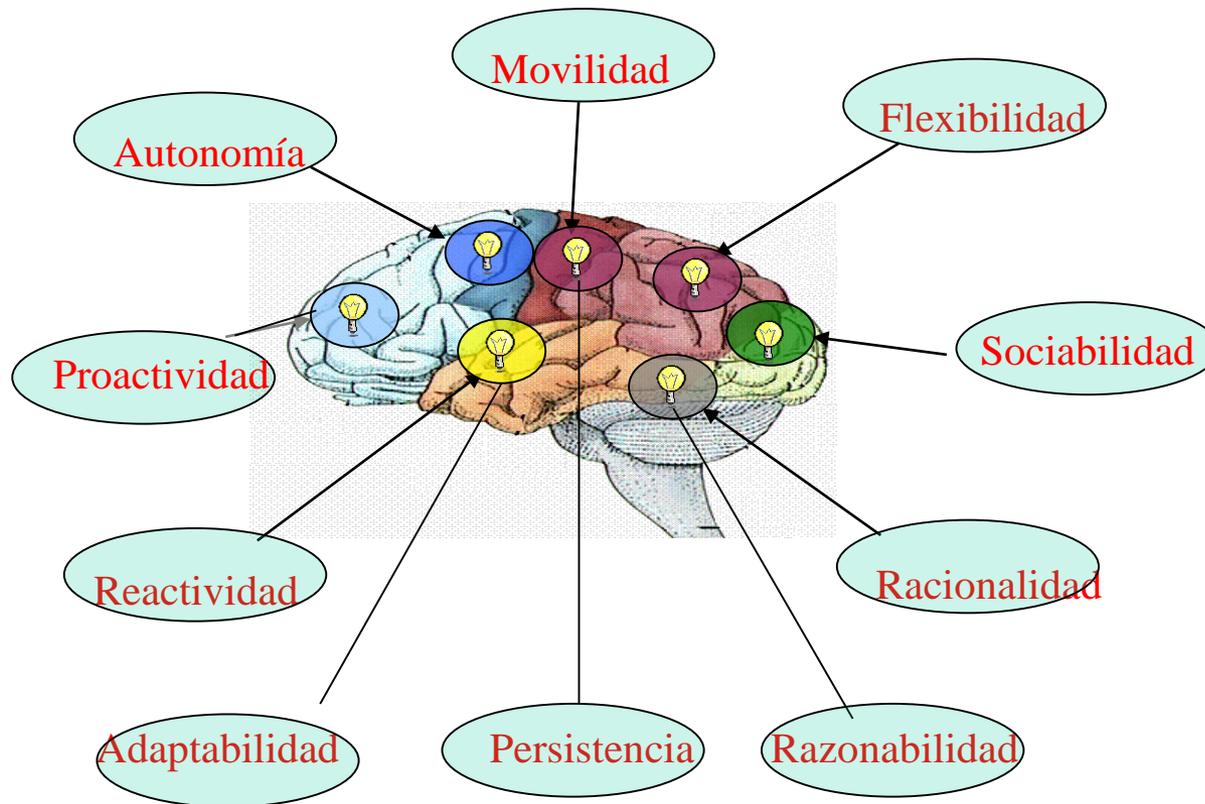
La inteligencia Artificial se ocupa del diseño del programa

Agentes



Agentes

Propiedades de un agente

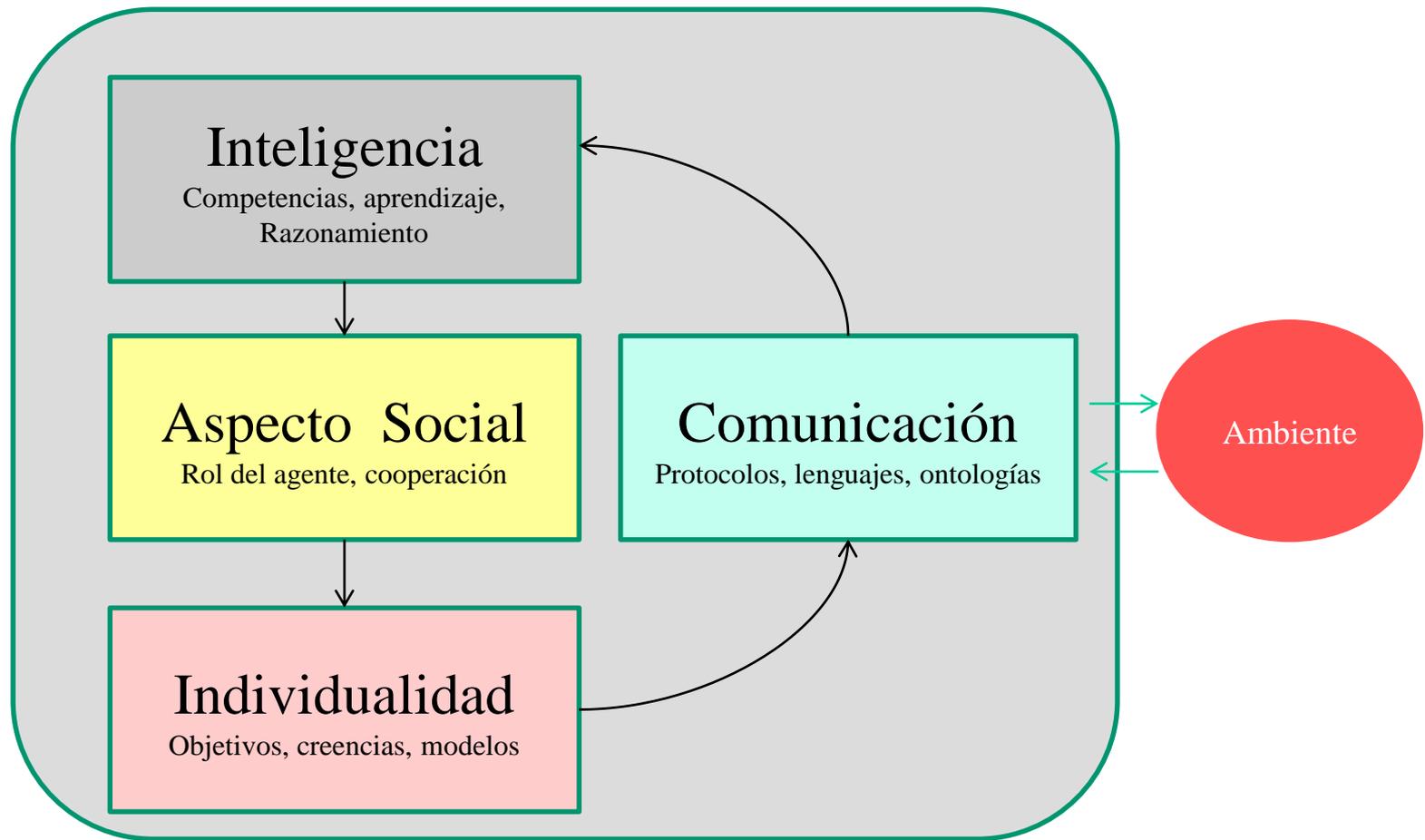


AGENTES

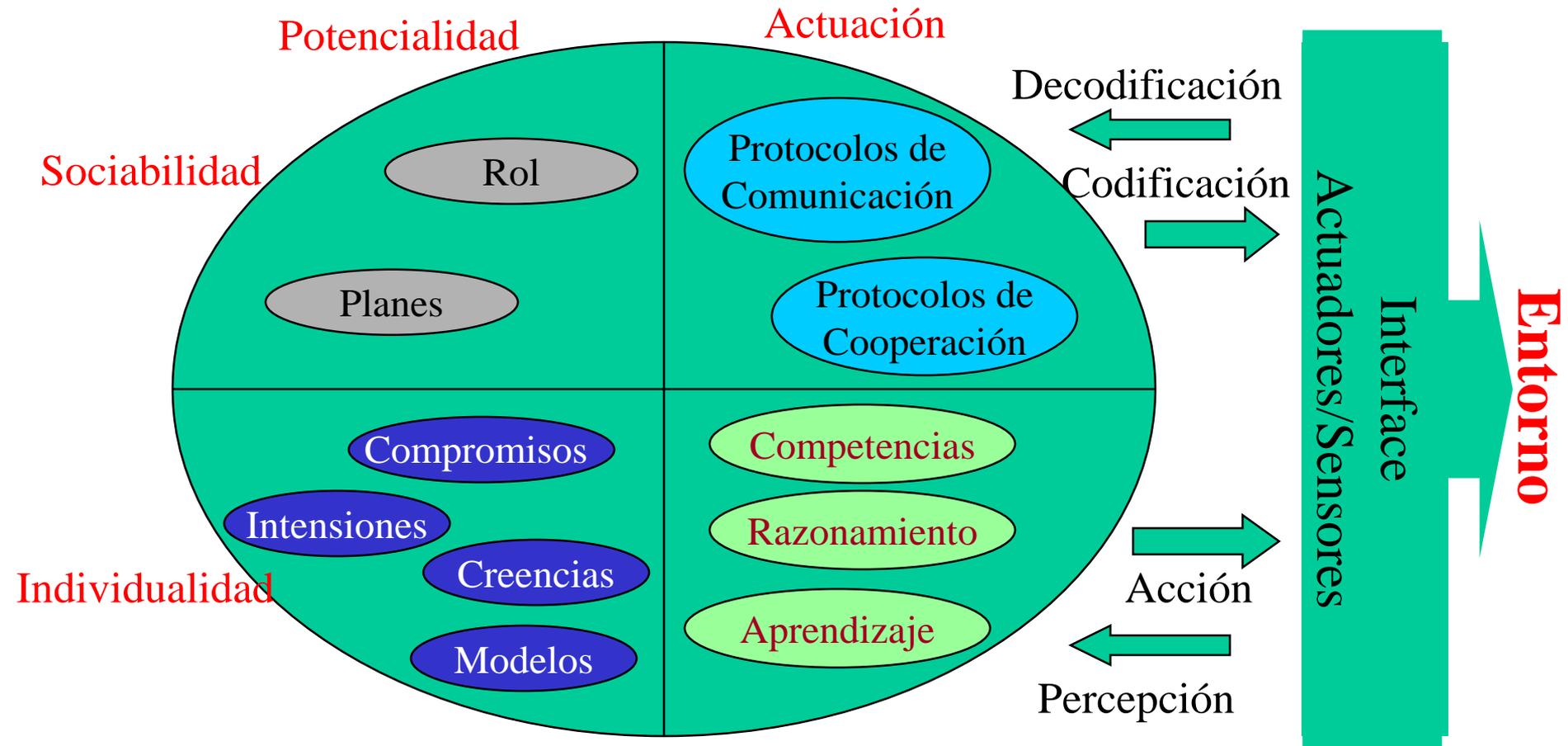
Propiedades

Autonomía	Comportamiento propio, y reacción en base a su estado interno
Adaptabilidad	Ajustarse a las necesidades del ambiente.
Reactividad	Percibir el entorno y responder a los cambios que ocurren en él
Proactividad	Reaccionar por iniciativa propia
Comunicación	Capacidad de cada agente de conversar utilizando un lenguaje basado en ontologías
Sociabilidad	Interactuar con otros agentes
Razonamiento	Decidir qué actividad realizar, cómo actuar, ordenar ideas
Racionalidad	Decidir cual acción seguir y en qué momento ejecutarla
Inteligencia	Analizar, ordenar ideas y conocimiento
Movilidad	Capacidad de un agente de trasladarse a través de una red telemática
Veracidad	Asunción de que un agente no comunica información falsa a propósito
Benevolencia	Un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos
Asincronismo	Un agente puede ejecutar sus acciones sin estar acoplado a otros agentes

Arquitectura de un Agente



Arquitectura de los Agentes



Agentes vs. Objetos

- Un Agente puede ser **autónomo**
- Un Agente es capaz de generar **comportamientos flexibles** (reactivos, proactivos)
- Un Agente puede **convivir en una comunidad** de ellos de forma racional (social)
- Desde el punto de vista de implantación, cada agente es un **elemento activo** (al menos un hilo de ejecución).

Procedimiento de base:

1. Percibo (*Actualiza Memoria*)
2. Decido (*Escoge Acción*)
3. Actúo (*Actualiza Memoria*)

Descripción práctica de un agente:

Sus Tareas.

Sus Conocimientos.

Su Comunicación



Otros aspectos importantes a considerar en los Agentes

- Mecanismos para **resolver un problema**
- Mecanismo para **planificar sus actividades** /tareas
- Mecanismos para **representar el conocimiento**
- Mecanismo de **razonamiento**
- Mecanismos de **aprendizaje**
- Mecanismos de **percepción**
- Mecanismos para **comunicarse**

Estructura Abstracta de los Agentes

- Estados del Ambiente

$$S = \{s_1, s_2, \dots\}$$

- Capacidades de actuar

$$A = \{a_1, a_2, \dots\}$$

- Agente Básico

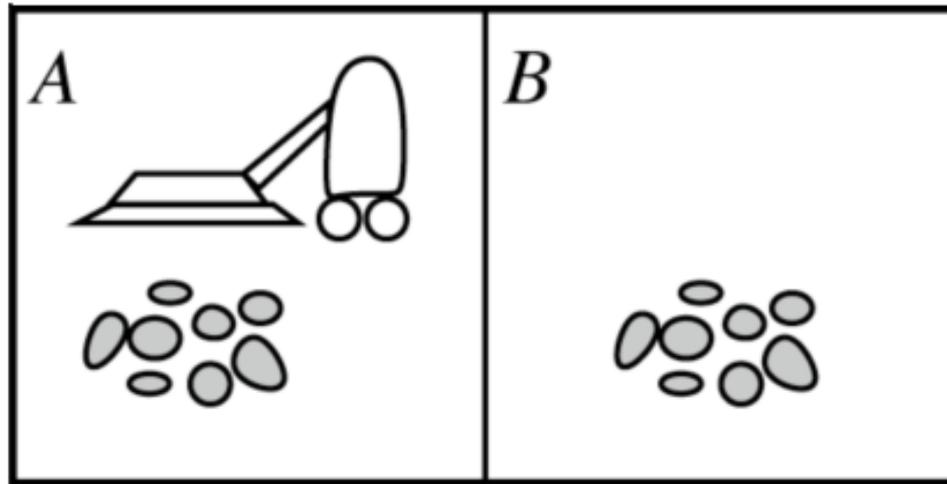
$$\text{acción: } S^* \rightarrow A$$

- Historia del Agente

$$h: s_0 - a_0 \rightarrow s_1 - a_1 \rightarrow s_2 \dots$$

Agente Aspiradora

- Percepciones: Localización y Contenido.
Por ejemplo [A, Sucio]
- Acciones: Izquierda, Derecha, Aspirar, Hacer_Nada



Agente Aspiradora

Secuencia de Percepciones	Acción
<i>[A, Limpio]</i>	<i>Derecha</i>
<i>[A, Sucio]</i>	<i>Aspirar</i>
<i>[B, Limpio]</i>	<i>Izquierda</i>
<i>[B, Sucio]</i>	<i>Aspirar</i>
<i>[A, limpio] [A, Limpio]</i>	<i>Derecha</i>
<i>[A, Limpio] [A, Sucio]</i>	<i>Aspirar</i>
<i>...</i>	<i>...</i>

Elementos de un Agente

- Medida de Rendimiento
- Secuencia de Percepción
- Conocimiento del ambiente
- Acciones que el agente realiza

mapeo de la secuencia percibida a las acciones

racionalidad: por cada secuencia percibida tomar acción basado en el conocimiento que tiene que maximice el rendimiento

Agentes Racionales

- Los agentes pueden ejecutar acciones con el objetivo de modificar percepciones futuras para obtener información útil (recopilación de información)
- El agente debe aprender lo máximo posible de lo que está percibiendo.
- Un agente es autónomo si su comportamiento queda determinado por su propia experiencia, compensando conocimiento incompleto o parcial

Agentes Racionales

- Un agente debe decidir hacer lo correcto, basándose en lo que puede percibir y las acciones que puede realizar.
- acción correcta = acción que lleva al agente a tener éxito en la realización de su tarea.
- ¿cómo evaluar esto? Medida de desempeño:

Ejemplo: las medidas de rendimiento del agente aspiradora pueden ser, entre otras:

- La cantidad de polvo eliminado
- La cantidad de electricidad consumida
- La cantidad de tiempo que tomó hacerlo

Mapeo Percepciones / Acciones

Un mapeo especifica la acción que un agente debe tomar en respuesta a una secuencia de percepciones.

- Puede ser en forma de tabla
 - Planes universales
 - Casos (indexados)
- Pueden ser en forma de funciones
 - Comportamiento estímulo/respuesta
 - Algoritmos de varios tipos

REAS

Las medidas de rendimiento, el entorno, los actuadores y los sensores (REAS) deben especificarse para guiar el diseño de agentes

- Ejemplo: la tarea de diseñar un vehículo autónomo
 - **Rendimiento:** seguro, rápido, viaje cómodo, maximizar ganancias.
 - **Entorno:** caminos, tráfico, peatones, clientes.
 - **Actuadores:** volante, acelerador, freno, señales, corneta
 - **Sensores:** cámaras, sonar, tacómetro, GPS, sensores en el motor.

Ejemplo descripción REAS

Sistema de diagnóstico Médico

- **Medida Rendimiento:** Salud paciente, mínimo costo
- **Ambiente:** Hospital, staff medico, paciente
- **Actuadores:** Sistema de preguntas, tratamientos,
- **Sensores:** Teclado, sensores de síntomas, respuestas de pacientes,

Ejemplo descripción REAS

- Sistema de la Torre de Control de un Aeropuerto,
- Sistema de Gestión de una empresa que hace calzados
 - Medida Rendimiento:
 - Ambiente:
 - Actuadores:
 - Sensores:

Ejemplo descripción

PASE/REAS

- Sistema de Gestión de una empresa que hace helados
- Sistema de Torre de Control de un Puerto Maritimo,
 - Medida Rendimiento:
 - Ambiente:
 - Actuadores:
 - Sensores:

Ambiente

- Completamente observable o no (Accesibilidad)
- Determinista o estocástico
- Episódico o secuencial (Interdependencia de eventos)
- Estático o no (Dinamismo)
- Discreto o Continuo (Continuidad)

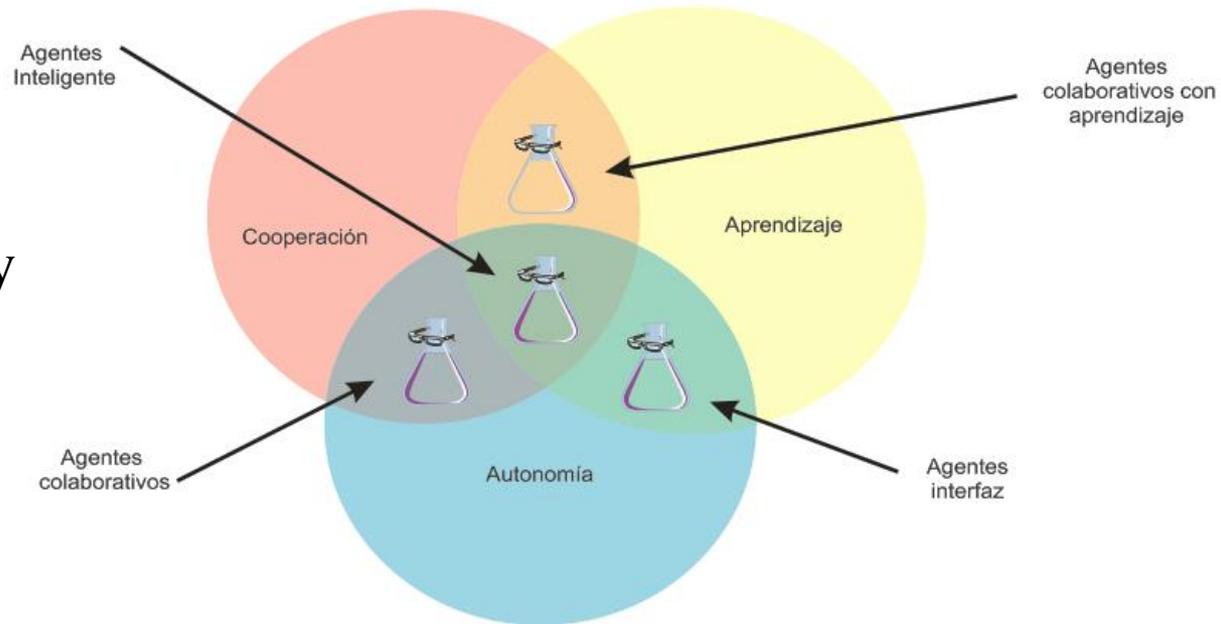
Ejemplos de Ambientes

	Ajedrez	Conducir
Compl. observable	Si	No
Determinista	Si	No
Episódico	No	No
Estático	Si	No
Discreto	Si	No

Tipos de Agentes

Muchas clasificaciones en la literatura

Según tres propiedades:
Autonomía, Aprendizaje y
Cooperación



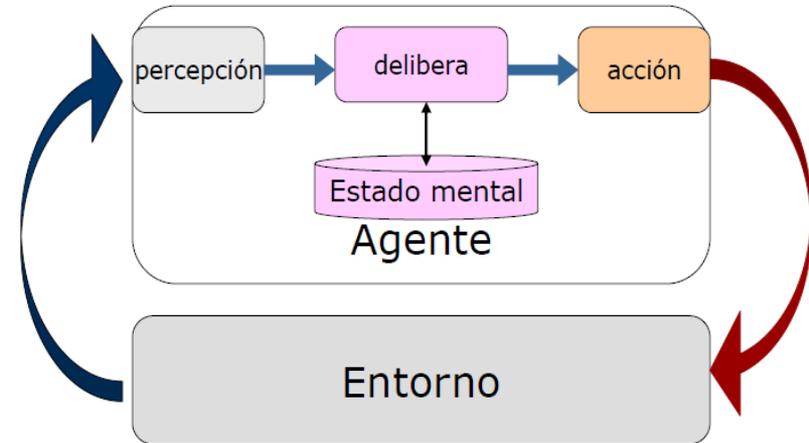
Tipos de Agentes

- **BASADOS EN CAPACIDADES:**
 - PURAMENTE COMUNICANTE: 1, 2, 3, 4, 6, 7, 9
 - PURAMENTE SITUADO: 1, 3, 4, 5, 6, 7, 8, 9
 - ...
- **BASADOS EN *MODOS DE CONDUCTA***
 - DIRIGIDA POR OBJETIVOS PRECISOS (OP)
 - DIRIGIDO POR LAS PERCEPCIONES (P)

Tipos de Arquitectura de un Agente

Arquitecturas Deliberativas

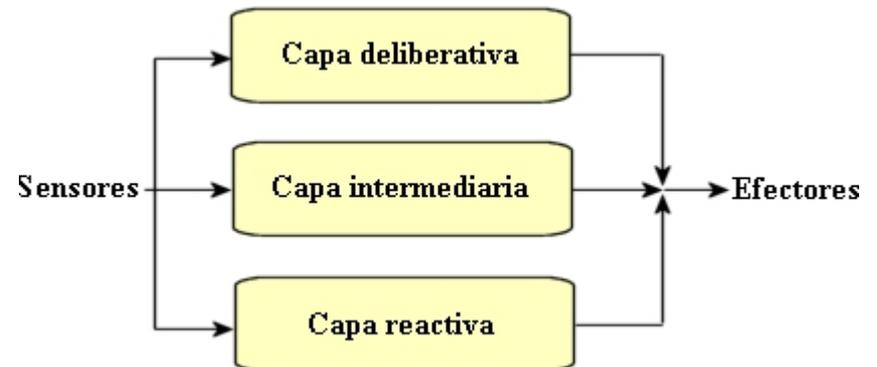
Contiene un modelo del mundo simbólico y explícitamente representado.
Se introduce una función deliberativa entre la percepción y la ejecución para elegir la acción correcta



Basados en deducción lógica

- ver: $S \rightarrow P$
- Acción: $D \rightarrow A$ D: BD de sentencias lógicas
- próximo: $DxP \rightarrow D$

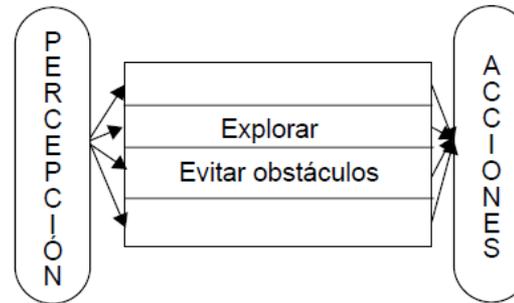
Arquitecturas Híbridas
Combinan lo mejor de las arquitecturas deliberativas y reactivas.
Puede estructurar en capas



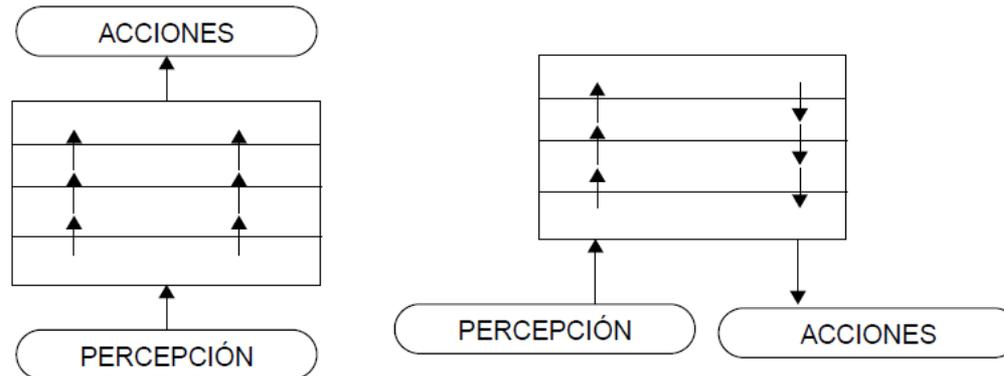
Tipos de Arquitectura de un Agente (según toma de decisión)

Arquitecturas por Capas

Capas horizontales:
todas las capas están
conectadas a la entrada y
salida del agente



• **Capas verticales:** la
entrada y la salida están
conectadas a una
única capa del agente



Tipos de Agentes (según toma de decisiones)

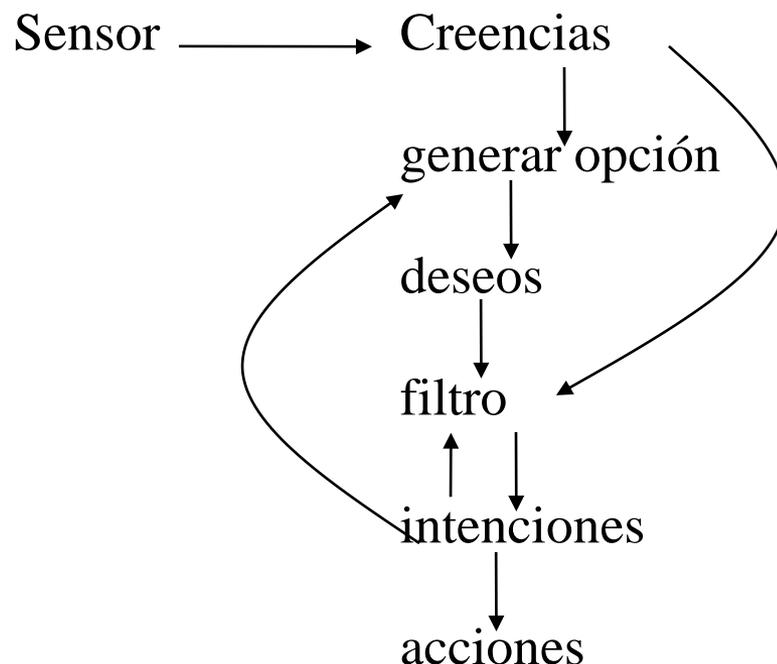
- **Arquitectura multinivel:** depende de la interacción entre varios elementos que razonan sobre distintas abstracciones del entorno



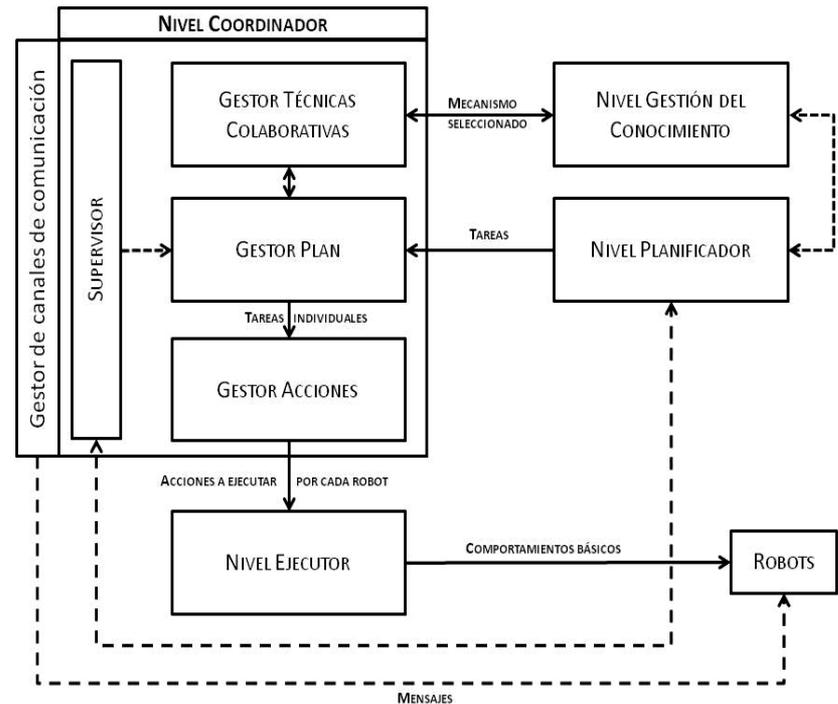
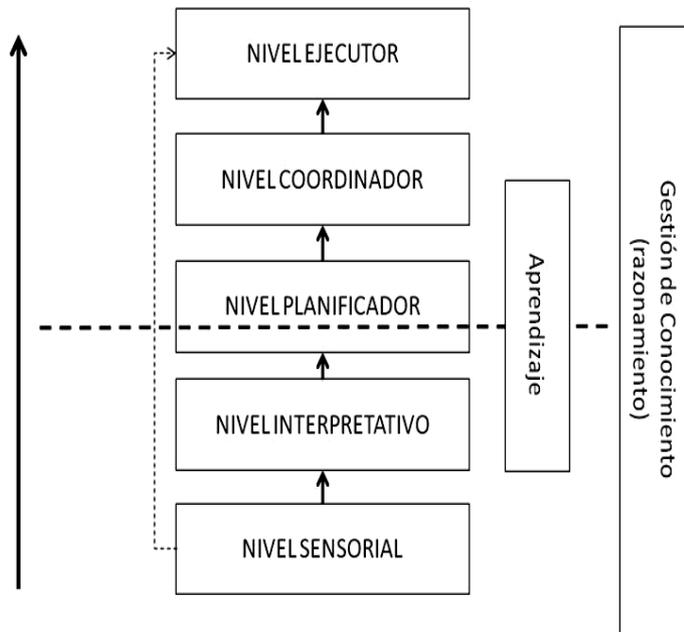
Subsistema de Control

Tipos de Agentes (según toma de decisiones)

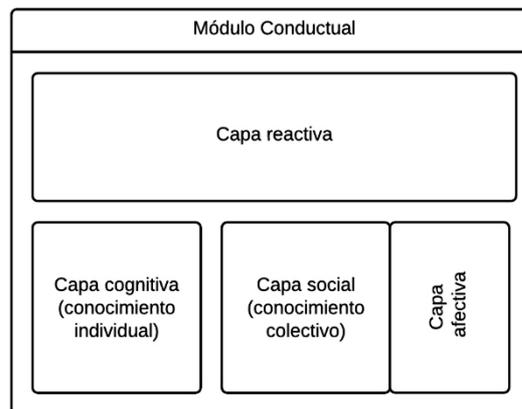
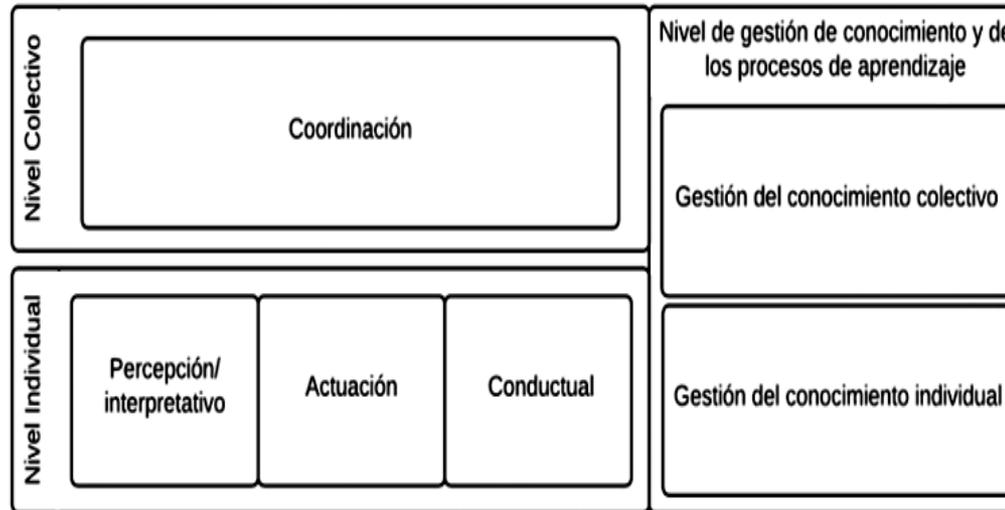
- **Agentes BDI** (*Belief, Desires and Intentions*) depende de la manipulación de estructuras de datos que representan las creencias, deseos e intenciones del agente
 - **Creencias:** Modelan el estado del Mundo.
 - **Deseos:** Permiten la elección de estados posibles del mundo.
 - **Intenciones:** Son el compromiso para alcanzar un cierto estado



Tipos de Agentes: Arquitectura multinivel



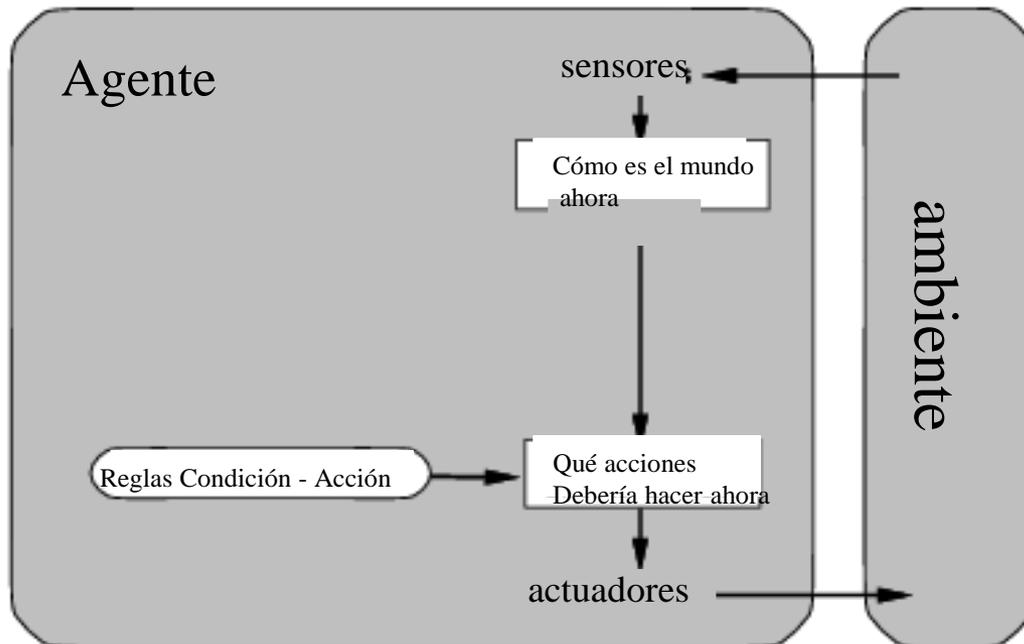
Tipos de Agentes: Arquitectura multinivel SMA



Tipos de Agentes (según relación entre percepción y acciones)

Agente Reflexivo

1. Interpreta (percepción)
2. Determina regla (si $\langle \text{---} \rangle$ entonces $\langle \text{---} \rangle$)
3. Actúa (según regla)



función AGENTE-REACTIVO-SIMPLE(percepción) devuelve una acción
estático: reglas, un conjunto de reglas condición-acción

```
estado ← INTERPRETAR-ENTRADA(percepción)
regla ← REGLA-COINCIDENCIA(estado, reglas)
acción ← REGLA-ACCIÓN[regla]
devolver acción
```

Seleccionan las acciones sobre la base de las percepciones actuales,

Por ejemplo:

Si el vehículo de delante frena y sus luces de freno se encienden
Entonces el agente taxista deberá advertirlo y frenar también

Se almacena de la forma:

Si vehículo delante frena entonces empezar a frenar

Los humanos tenemos muchas de estas asociaciones:

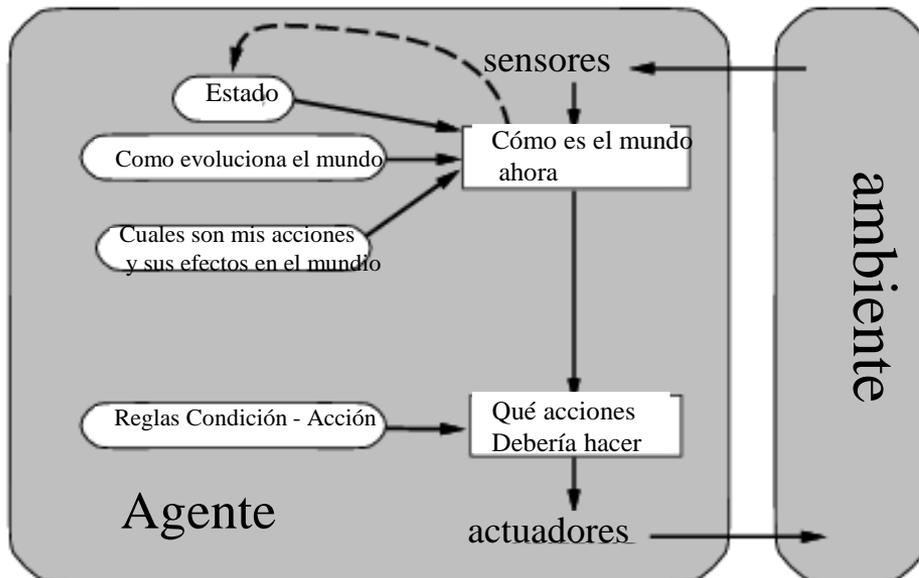
Reflejos innatos
Respuestas aprendidas

Tipos de Agentes (según relación entre percepción y acciones)

Agente Reflexivo con Estado Interno

1. Actualiza estado (estado, percepción)
2. Determina regla (estado, reglas)
3. Actúa (según regla)

Requiere de modelos del mundo!!



función AGENTE-REACTIVO-CON-ESTADO(percepción) devuelve una acción

estático: *estado*, una descripción actual del estado del mundo

reglas, un conjunto de reglas condición-acción

acción, la acción más reciente, inicialmente ninguna

estado ← ACTUALIZAR-ESTADO(*estado*, *acción*, *percepción*)

regla ← REGLA-COINCIDENCIA(*estado*, *reglas*)

acción ← REGLA-ACCIÓN[*regla*]

devolver *acción*

La percepción actual se interpreta a partir del estado anterior utilizando información sobre:

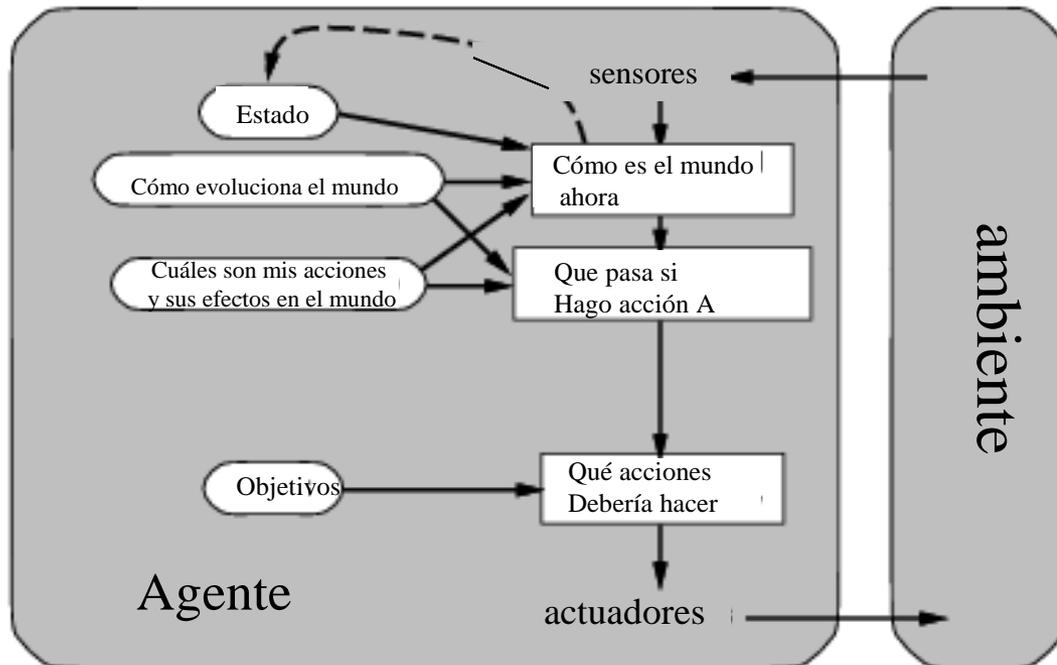
- Cómo evoluciona el mundo independientemente del agente
- Cómo influyen en el mundo las acciones del agente

Por ejemplo:

Si el vehículo de delante es un modelo antiguo no se podrá determinar si ha frenado a partir de una única imagen.

Agente con Objetivos

1. Actualiza Memoria (estado, según lo que percibió)
2. Determina acción (estado, qué yo quisiera cumplir (objetivos): *búsqueda y/o planificación*)
3. Actúa (Acción inmediata, o la que corresponda según la secuencia obtenida por el proceso de búsqueda y/o plan.)
4. Actualiza estado (estado, acción)



Un agente basado en objetivos utiliza una descripción de las metas a alcanzar que le sirve para escoger entre distintas acciones posibles.

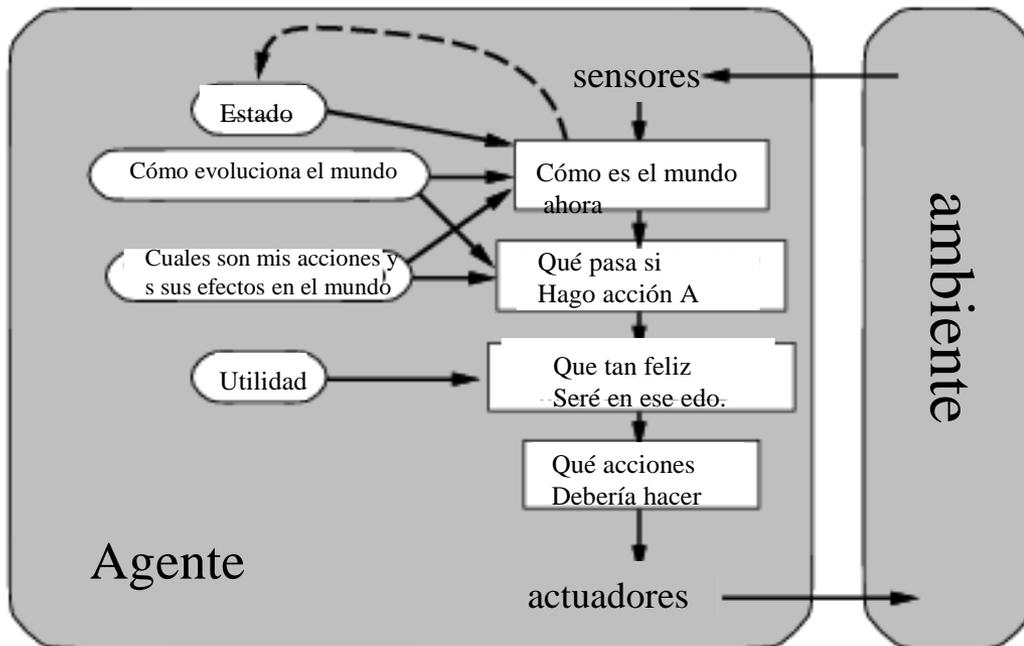
Por ejemplo:
El destino del pasajero

La búsqueda y a planificación son dos subáreas de la Inteligencia Artificial dedicadas a la resolución de problemas utilizando objetivos

juegos es una subárea de la Inteligencia Artificial que utiliza funciones de utilidad en la resolución de problemas

Agente con Función de Utilidad

1. Actualiza Memoria (estado, según lo que percibió)
2. Determina acciones (estado, qué yo quisiera cumplir (objetivos))
3. Selecciona acción (acciones, función de utilidad)
4. Actúa (según acción seleccionada)
5. Actualiza estado (estado, acción)



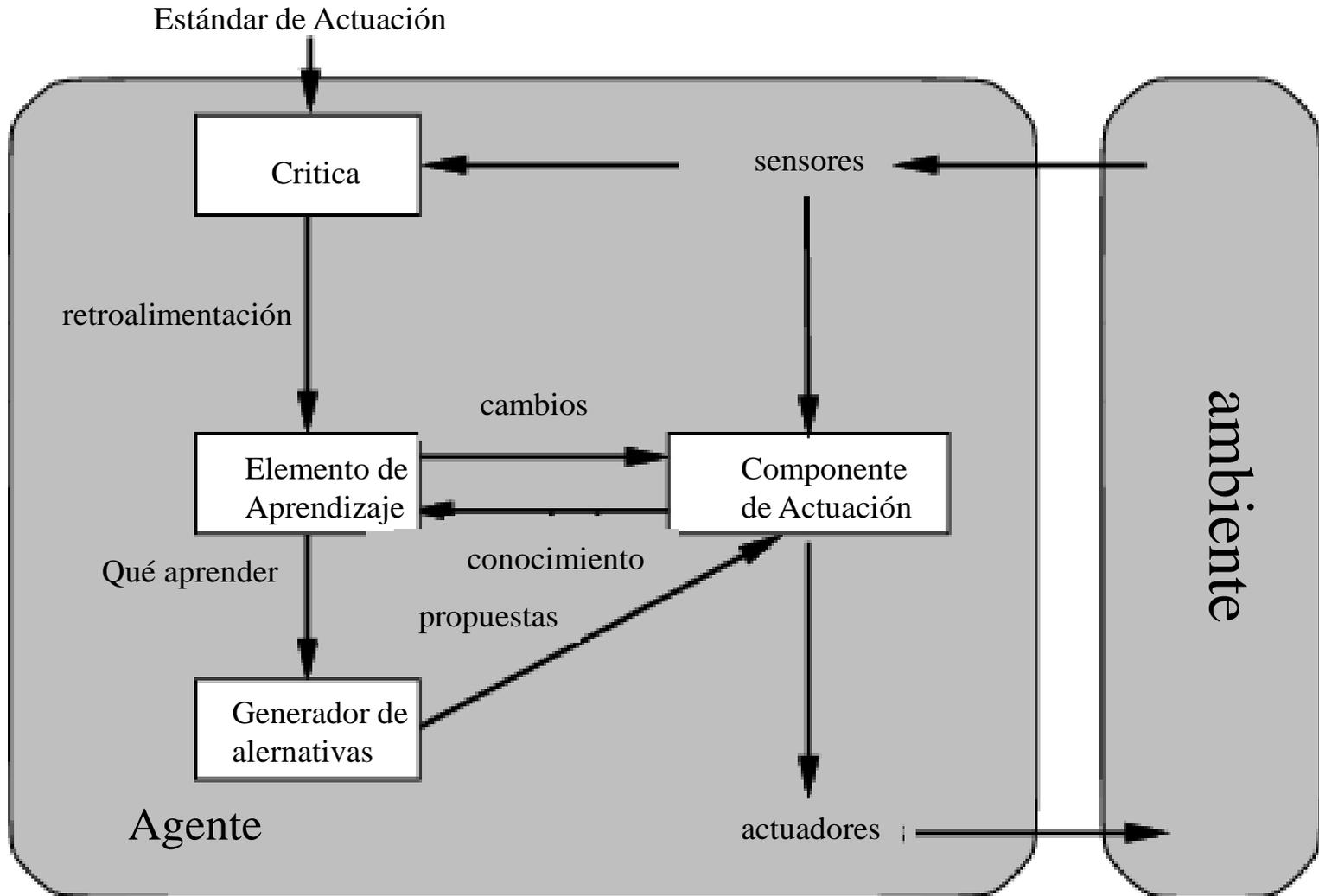
Los objetivos únicamente no bastan para asegurar un comportamiento ideal.

Por ejemplo:

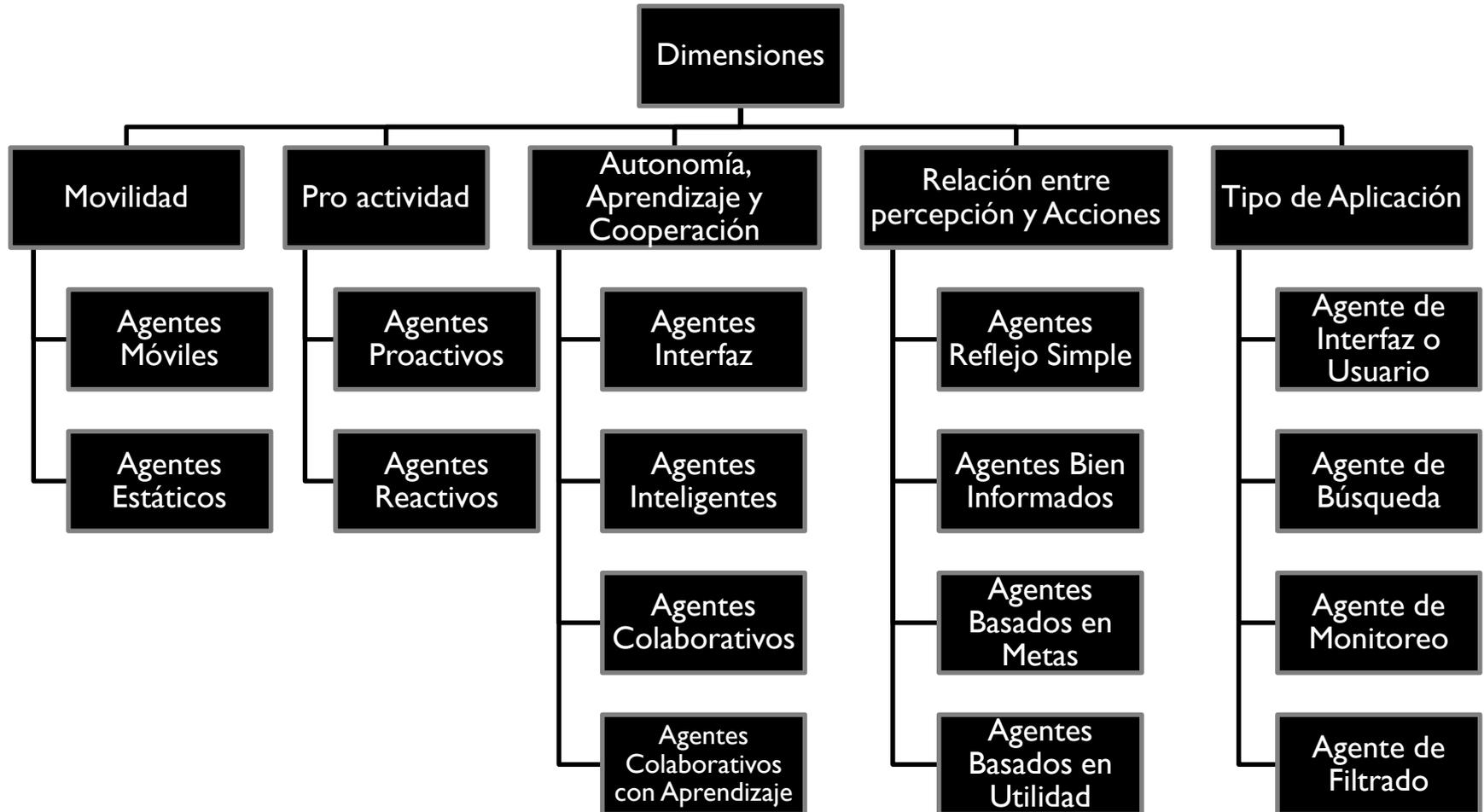
Diferentes secuencias de acciones pueden llevar al destino que el pasajero desea, pero algunas son más rápidas, seguras o baratas que otras.

Un agente basado en la utilidad usa un criterio para estimar el grado de satisfacción de un estado para el agente que le sirve para escoger entre distintas acciones validas

Agente que Aprende



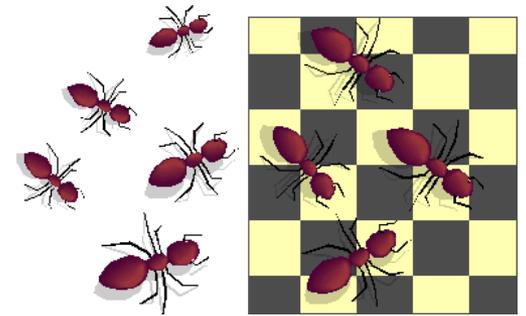
Tipos de Agente



Identifique el tipo de agente para cada caso

- Agente supervisor de una planta industrial
- Agente que conduce un automóvil
- Agente asistente de un médico en tareas de diagnóstico
- Agente controlador de un aeropuerto

Soluciones de problemas mediante agentes de búsqueda



Jose Aguilar

Cemisid, Facultad de Ingeniería

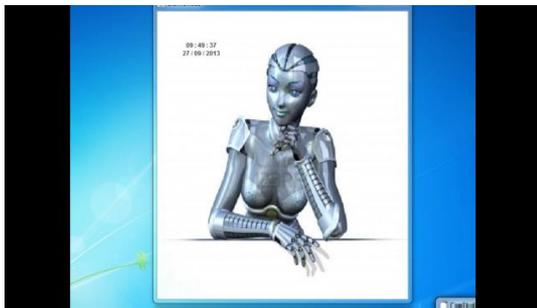
Universidad de los Andes

Mérida, Venezuela

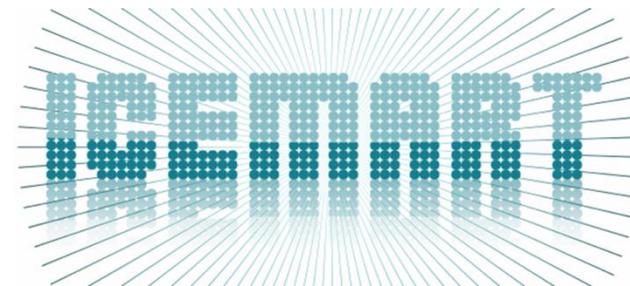
aguilar@ula.ve

Icemart: La Nevera Inteligente

Objetivo: ayuda a gestionar la alimentación del usuario, ofreciendo distintos servicios entre los cuales está la creación de dietas basadas en su organismo, recomendación de platos y creación de listas de compras con los alimentos escasos en la nevera



J. Aguilar



Agentes para la solución de problemas

Son agentes basados en metas que determinan que deberán hacer por medio de secuencias de acciones que les permitan obtener estados deseables.

Pasos para la solución de problemas:

- 1. Formulación de objetivos/metas:** se establece el objetivo
- 2. Formulación del problema:** se decide que acciones y estados habrán de considerarse.
- 3. Búsqueda:** evaluación de las posibles secuencias de acciones que le llevan a la meta y elección de la más apta.
- 4. Ejecución:** se llevan adelante la solución que presenta la búsqueda.

Agentes y Búsqueda

Partes

- Formulación del Objetivo (*edo. actual, medida de rend.*)
- Formulación del Problema (*acciones y edos. a considerar*)
- Algoritmo de Búsqueda de Solución
- Ejecutar

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
  static: seq, an action sequence, initially empty
           state, some description of the current world state
           goal, a goal, initially null
           problem, a problem formulation

  state ← UPDATE-STATE(state, percept)
  if seq is empty then do
    goal ← FORMULATE-GOAL(state)
    problem ← FORMULATE-PROBLEM(state, goal)
    seq ← SEARCH(problem)
  action ← FIRST(seq)
  seq ← REST(seq)
  return action
```

Formulación de un problema

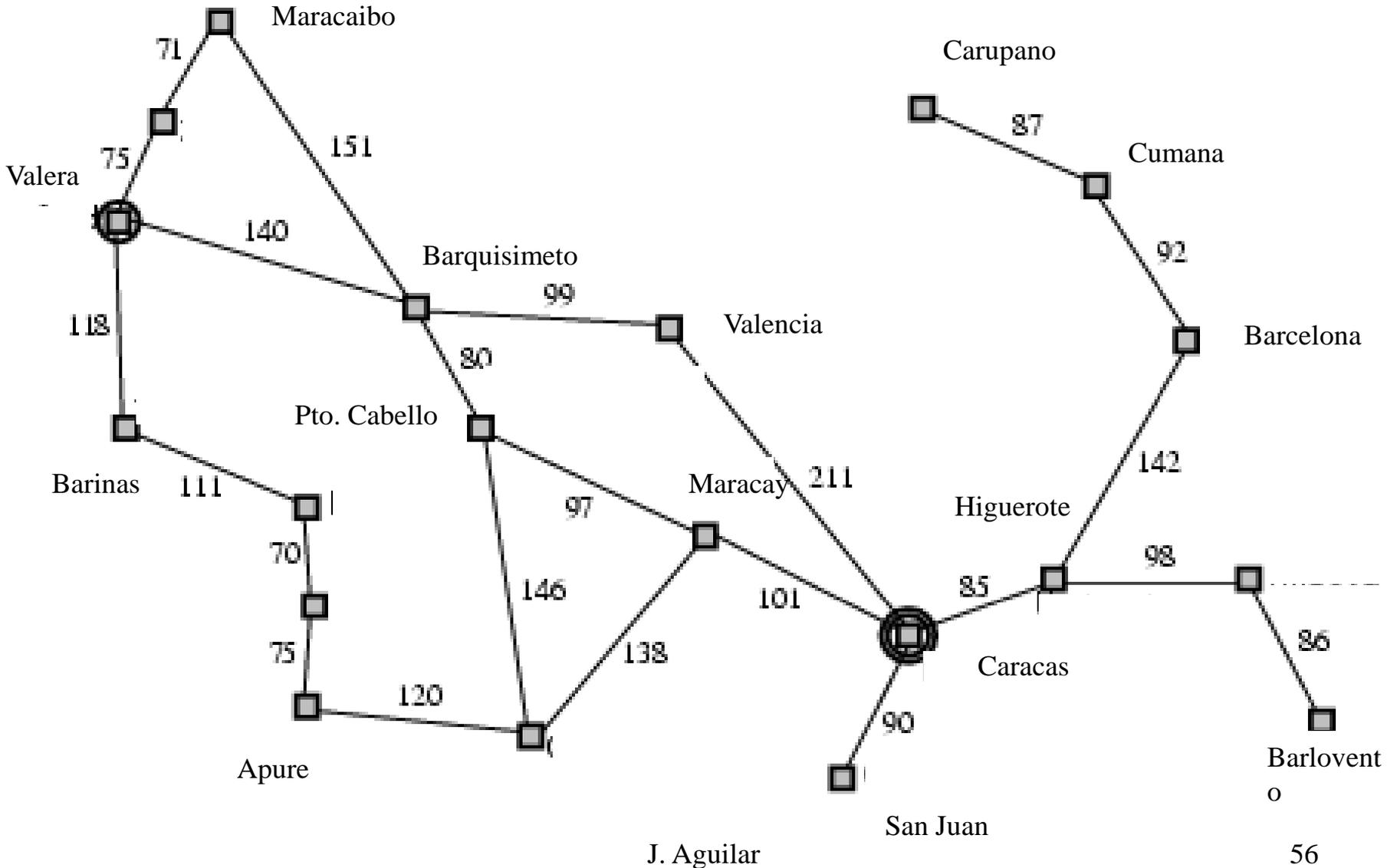
es el procesos que consiste en decidir *qué acciones y estados* habrán de considerarse y es el paso que sigue a la formulación de metas.

- Para el problema de viajar a través de Venezuela, se considera **un estado** el estar en una ciudad determinada, y como **acción** el traslado de una ciudad a otra.

Ejemplo: Viajar

- Estoy en Valera y quiero ir a Caracas
- **Objetivo:**
 - Estar en Caracas
 - **Estado Inicial:** Valera
- **Problema:**
 - **Espacio de Estados (camino):** recorrido por varias ciudades
 - **Acciones (Operadores):** ir entre las ciudades
- **Solución:**
 - **Secuencias de ciudades**, p.e., Valera, Barquisimeto, Valencia, Maracay, Caracas

Ejemplo: Venezuela



Ejemplo: Biblioteca

- Estoy en la biblioteca y necesito encontrar un libro de algebra cuyo autor es Boole
- **Objetivo:**
 - **Estado Inicial:**
- **Problema:**
 - **Espacio de Estados (camino):**
 - **Acciones (Operadores):**
- **Solución:**

Tipos de problemas

- **Determinista, todo observable** → **estado simple**
 - Agente conoce en que estado esta y estará, ya que sabe lo que hace cada acción. Solución es una secuencia
 - **Problemas de un solo estado:** el agente conoce con exactitud en que estado se encuentra y el resultado de cada una de sus acciones.

- **No-observable** → **no se percibe, múltiples estados**
 - Agente no tiene idea donde esta, pero sabe que hacen sus acciones.
 - Debe razonar sobre posibles estados que puede tener; solución: una secuencia
 - **Problemas de estados múltiples:** el agente no conoce con exactitud en que estado se encuentra, pero si el resultado de cada una de sus acciones.

Tipos de problemas

- No-determinista y/o parcial observable → **contingencia**
 - Percepción de **nueva** información sobre estado actual. solución es un árbol de acciones y ejecutar una rama es una conting.
 - Mecanismo de **intercalado** (buscar y ejecutar)
 - **Problemas de contingencias:** el agente no conoce con exactitud en que estado se encuentra, pero si el resultado de cada una de sus acciones, aunque se le pueden presentar ciertas contingencias en las mismas.
- Desconocido espacio de estados → **exploración**
 - No se conoce efecto de sus acciones
 - **Problemas de exploración:** el agente no conoce con exactitud en que estado se encuentra, ni el resultado exacto de cada una de sus acciones.

Mundo de la aspiradora

- En este mundo hay dos posibles ubicaciones.
 - En ellas puede o no haber mugre.
 - El agente se encuentra en una de las dos
- Son tres las acciones posibles: *A la izquierda*, *A la derecha* y *Aspirar*.
- Se puede suponer que la eficiencia del aspirado es 100%
- La meta es eliminar todo el sucio.

Agente Limpieza

1. Estado simple, inicial #5.

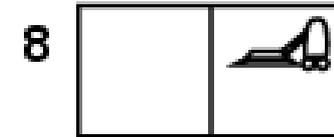
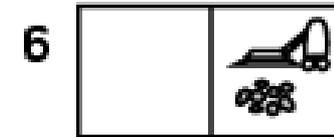
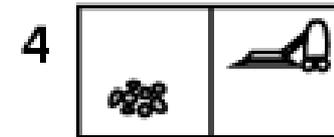
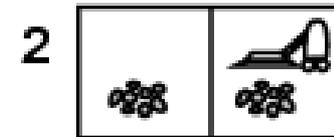
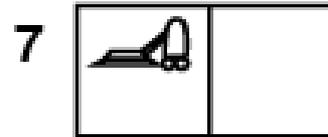
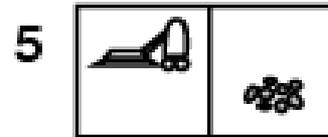
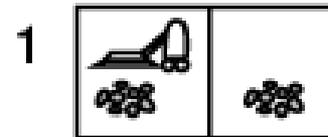
- Solución?
- agente debe decidir sobre los estados a ejecutar
- Una secuencia [*Der, Aspirar*]

2. Sin percibir, inicial Solución?

Una secuencia

[*Aspirar; Der, Aspirar*]

Conoce todo: estado
y efecto acción



No conoce estado

Agente Limpieza

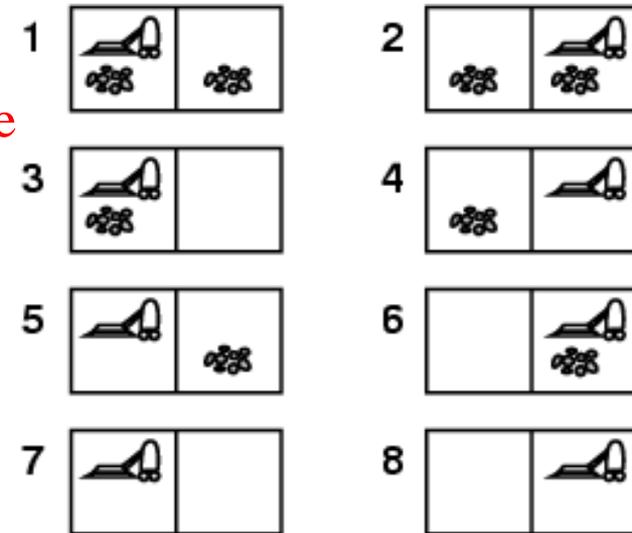
3. Contingencia

No conoce estado, pero percibe

- Parcialmente Observable

- Percepción: [Estar en #5 o #7]

Solución?

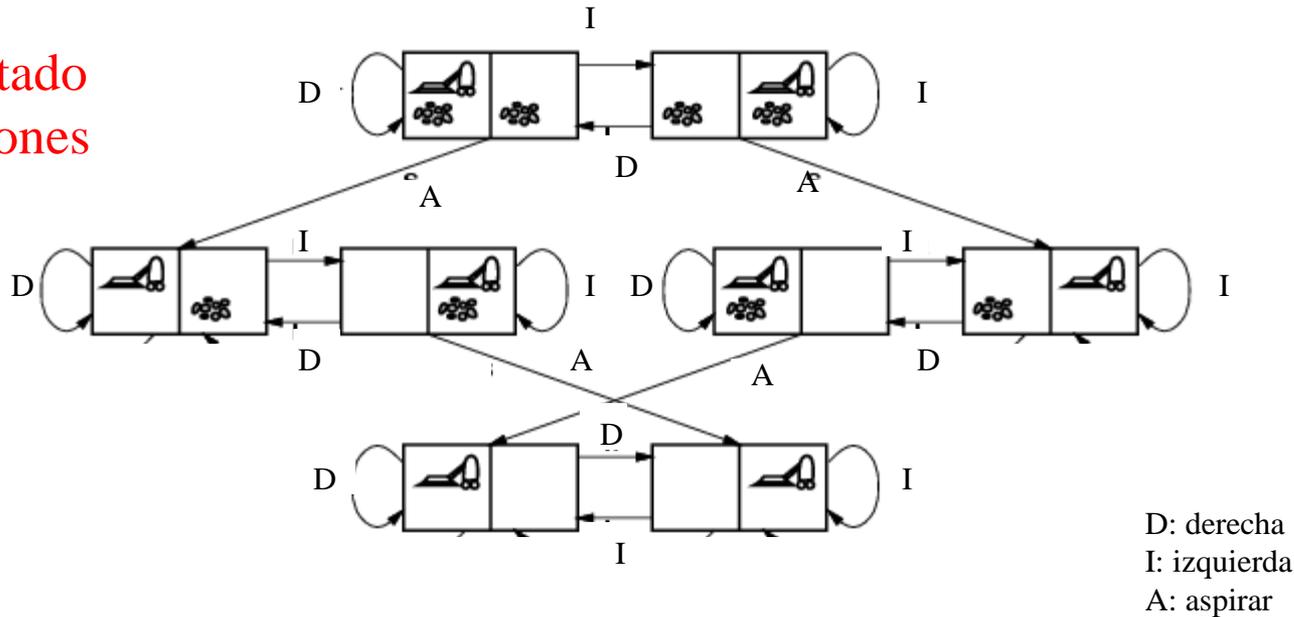


- Solución forma de árbol, donde cada nodo puede ser seleccionado según la **percepción** recibida en ese punto del árbol.

- [Der, si sucio entonces aspiro]

Agente Limpieza: exploración

No conoce estado
ni efecto acciones



- Estado? Localidad robot y limpio o no
- Acciones? Derecha, izquierda o aspirar
- Objetivos? Limpio en todas las localidades
- Costo? 1 por acción

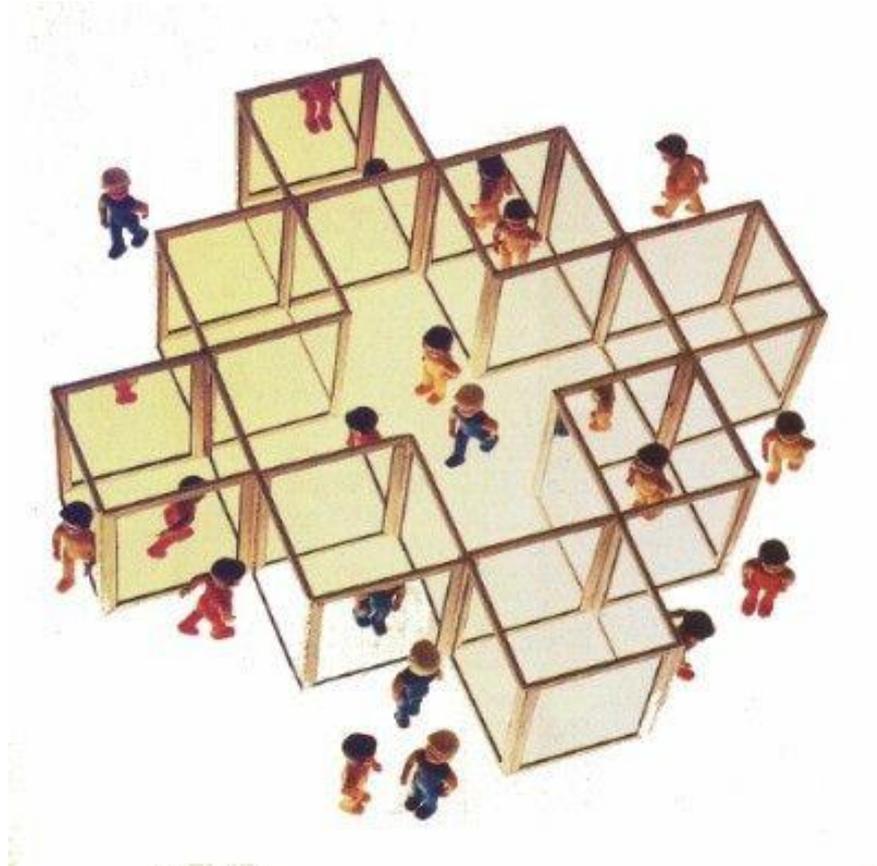
Agentes de Búsqueda

- **Definición de un problema desde la IA:**
 - Test Objetivo
 - Espacio de Estados
 - Estado Inicial
 - Operadores: Funcion sucesor
 - Función de Costo para evaluar caminos
- Otros aspectos:
 - Reducción del espacio

da caminos

Estado Inicial

Es el estado que el agente sabe que se encuentra.



Operadores

Posibles acciones que el agente puede emprender

- La formulación mas común usa una **función de sucesor**:

Dado un estado particular x , $S(x)$ regresa un conjunto de pares ordenados $\langle \text{acción}, \text{sucesor} \rangle$

$$S(\text{En}(\text{Mda})) \rightarrow \{ \langle \text{Ir}_a(\text{Bqto}), \text{En}(\text{Bqto}) \rangle, \\ \langle \text{Ir}_a(\text{Vra}), \text{En}(\text{Vra}) \rangle \}$$

Espacio de Estados

Conjunto de todos los estados que pueden alcanzarse a partir del estado inicial, mediante una secuencia de operadores.

- La función sucesor define el **espacio de estados**,
- Una **ruta** en el espacio de estados es una secuencia de acciones que permite pasar de un estado a otro.

Test objetivo

El test objetivo es lo que el agente aplica para decidir si se trata de un estado meta.

- A veces se compara el estado actual con un conjunto de estados meta,
- En otros, se busca una propiedad en el sistema.



Función de costo

Función mediante la que se asigna un costo a una solución dada.

- El costo total es la suma de todos los costos de cada una de las acciones individuales del agente.

Solución

- Los **cuatro elementos indicados** (estado inicial, operadores, prueba de meta y función costo de ruta) serán las **entradas al algoritmo de búsqueda.**
- **La salida** producida por el algoritmo es **la solución,**

Por ejemplo, una ruta que va desde el estado inicial al estado que satisface el test objetivo.

Mundo de la aspiradora

Estado inicial:

Cualquier estado puede ser designado como estado inicial.

Operador:

Genera los estados legales que resultan de intentar las tres acciones (*izquierda, derecha y aspirar*).

Test objetivo:

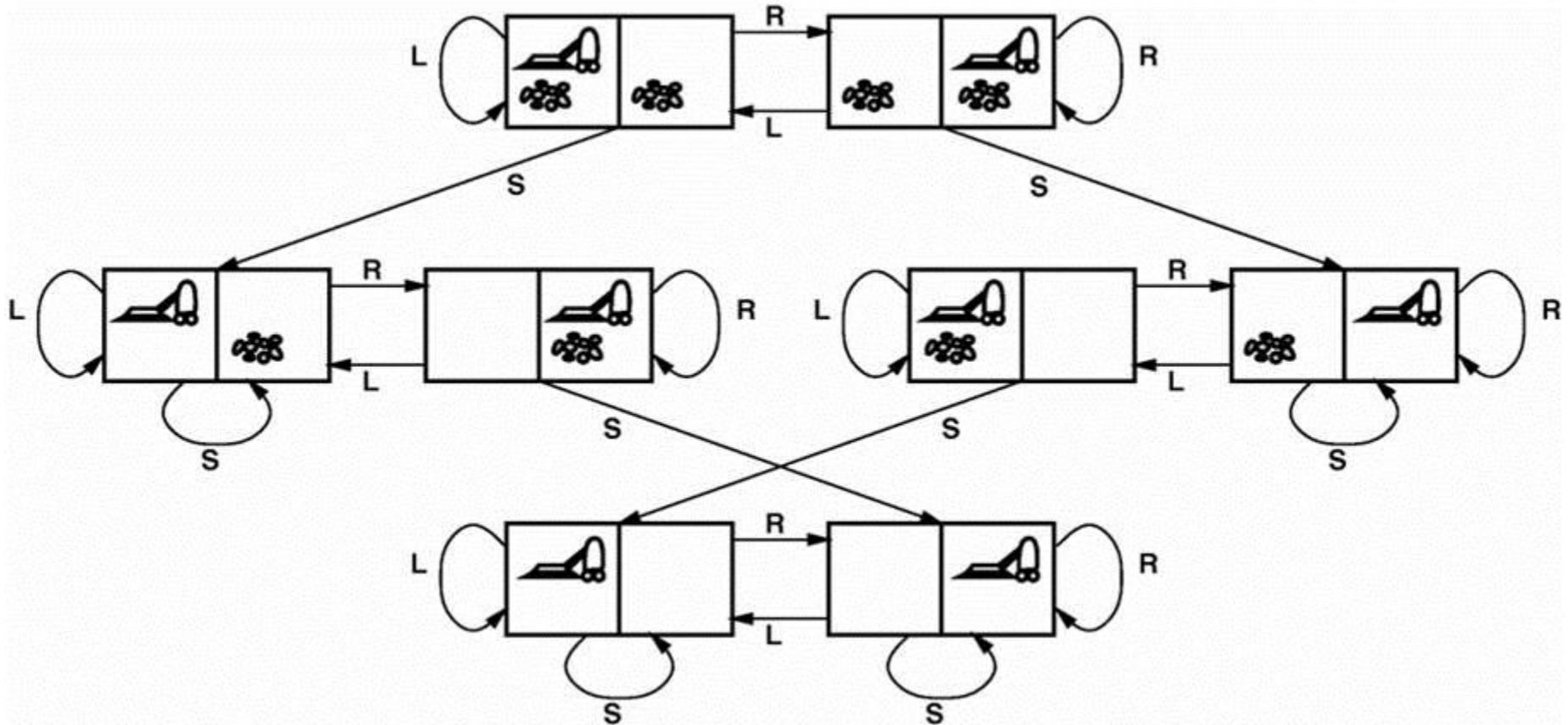
Revisa si las dos ubicaciones están limpias.

Función de Costo:

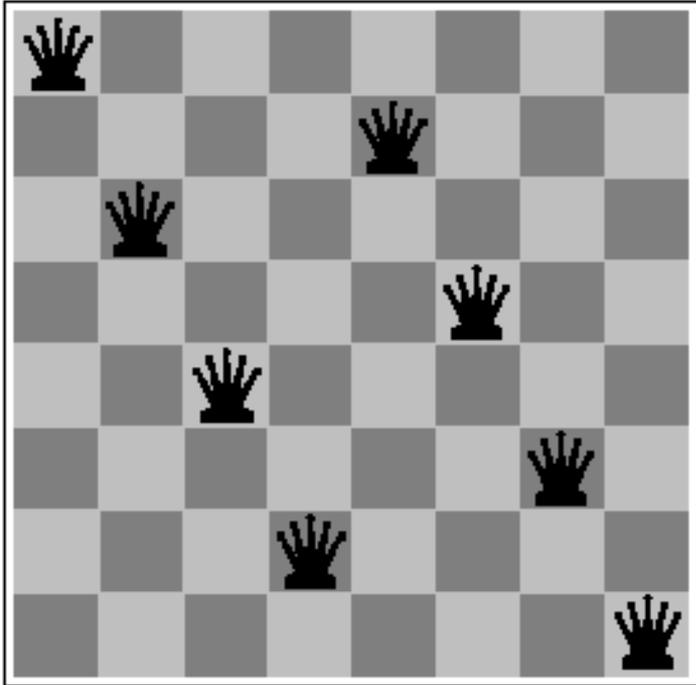
Cada paso cuesta 1, así el costo de ruta es el número de pasos en la ruta.

Mundo de la aspiradora

Diagrama de Estados



El problema de las ocho reinas



Colocar 8 reinas en un tablero de ajedrez de manera que ninguna de ellas esté en posibilidad de atacarse entre sí

Estados:

Cualquier disposición que tenga de 0 a 8 reinas en el tablero.

El problema de las ocho reinas

Estado inicial:

ninguna reina en el tablero.

Función de sucesor:

añadir una reina a un cuadro vacío.

Test objetivo:

8 reinas en el tablero, ninguna con posibilidad de atacar.

Función de Costo:

cero

El problema de las ocho reinas

Otras Formulaciones

- **Formulación 2**
 - **Estados:** Disposiciones de 0 a 8 reinas, sin que se pueda atacar ninguna de ellas
 - **Función de Sucesor:** Poner una reina en la columna vacía del extremo izquierdo de manera que ninguna otra reina la pueda atacar.
- **Formulación 3**
 - **Estados:** Disposiciones de 8 reinas, una en cada columna.
 - **Función de sucesor:** Cambiar las reinas que puedan ser atacadas a otro cuadro de la misma columna.

Determinación de una ruta aérea

- Consiste en definir una ruta **en función de las ubicaciones y las transiciones** mediante los vínculos que relacionan una y otra ubicación.
- Los algoritmos que resuelven estos problemas se usan en:
 - ruteo en redes de comunicación,
 - sistemas para planificación de viajes aéreos,

Determinación de una ruta aérea

- **Estados:** Cada uno es representado por un aeropuerto y la hora actual.
- **Estado inicial:** Es especificado por el problema.
- **Función de sucesor:** Regresa los estados resultantes de tomar cualquier vuelo programado, que sale después de la hora actual, más el tiempo de vuelo entre aeropuertos, desde el aeropuerto actual a otro.
- **Test objetivo:** Verifica si estamos en el destino a una hora pre-especificada.
- **Función de Costo:** Depende del costo monetario, el tiempo de espera, el tiempo de vuelo, procedimientos de migración, calidad de asiento, hora del día, tipo de avión, bonificaciones por viajes frecuentes, etc.

Determinación de una ruta terrestre

- Estado Inicial: Valera
- Objetivos
 - **Explicito:** $x = \text{“ir a Caracas”}$
- Función de Costo
 - Suma de distancias,
- Espacio de Estados (camino)
- Operador $S(x) = \text{accion-nuevo estado}$
 - $S(\text{Valera}) = \{ \langle \text{Valera} \rightarrow \text{Bqto}, \text{Bqto} \rangle, \dots \}$

Hay casos donde el objetivo es Implícito
Jaquemate(x)

Búsqueda de Soluciones

Después de definir un problema, se prosigue con la búsqueda de la solución.

- La búsqueda se hace en el espacio de estados, y la idea es mantener y ampliar un conjunto de secuencias de solución parciales.
- Se busca generar una secuencia de acciones para lograr el estado meta.

Agentes y Búsqueda

- **Algoritmo de búsqueda:**
 - Permitir la transición entre estados usando los operadores
 - Controlar esos movimientos
- **Proceso de Búsqueda**
 - **búsqueda ciega (Búsquedas sin contar con información):** no utiliza información sobre el problema. No existe información acerca de los pasos necesarios o costos para pasar de un estado a otro. Normalmente, se realiza una búsqueda exhaustiva.
 - **Búsqueda heurísticas (Búsqueda respaldada con información):** usan información sobre el problema como costos, etc. se posee información muy valiosa para orientar la búsqueda para que sea mas óptima