



Universidad  
Francisco de Vitoria  
**UFV** Madrid

# *Ingeniería del Conocimiento*

---

## **Tema 9: Sistemas Basados en el Conocimiento**

# Objetivos del tema



- Ubicación
  - Unidad 3: **EL CONOCIMIENTO Y SU REPRESENTACION**
  - *Tema 9: Sistemas Basados en el Conocimiento*
  
- Objetivos generales
  - Saber qué es un **Sistema Basado en el Conocimiento**
  - Entender la **estructura** de un SBC **independientemente** del formalismo de representación y del mecanismo de razonamiento.
  - Aplicar este conocimiento al caso de los **Sistemas de Producción**.
  - Ver el desarrollo de un SBC como un **problema de ingeniería**, con sus técnicas, metodologías y equipos de trabajo.



1. Introducción
2. Sistemas Basados en el Conocimiento
3. Estructura de un SBC
4. Ingeniería del Conocimiento
  1. IC e ISW
  2. Ciclo de Vida de un SBC
  3. Roles del equipo



1. **Introducción**
2. Sistemas basados en el Conocimiento
3. Estructura de un SBC
4. Ingeniería del Conocimiento
  1. IC e ISW
  2. Ciclo de Vida de un SBC
  3. Roles del equipo

# 1. Introducción



- Los *SBC* surgen en los años 70, con los sistemas expertos
  - Extracción del conocimiento especializado (a partir de expertos humanos, libros, etc.) y representación en bases de conocimiento
  
- La *Ingeniería del Conocimiento* nace a finales de los 80 (*crisis de los SBC*)
  - Proceso de desarrollo de un SBC
  - El “desarrollador” de un SBC (KBS) se llama *Ingeniero del Conocimiento*
    - Sus conocimientos incluyen
      - Reconocer qué conocimiento se utiliza para resolver un problema
      - Determinar cuál es la mejor manera de representarlo
      - Ser capaz de desarrollar una herramienta adecuada si no la hubiera



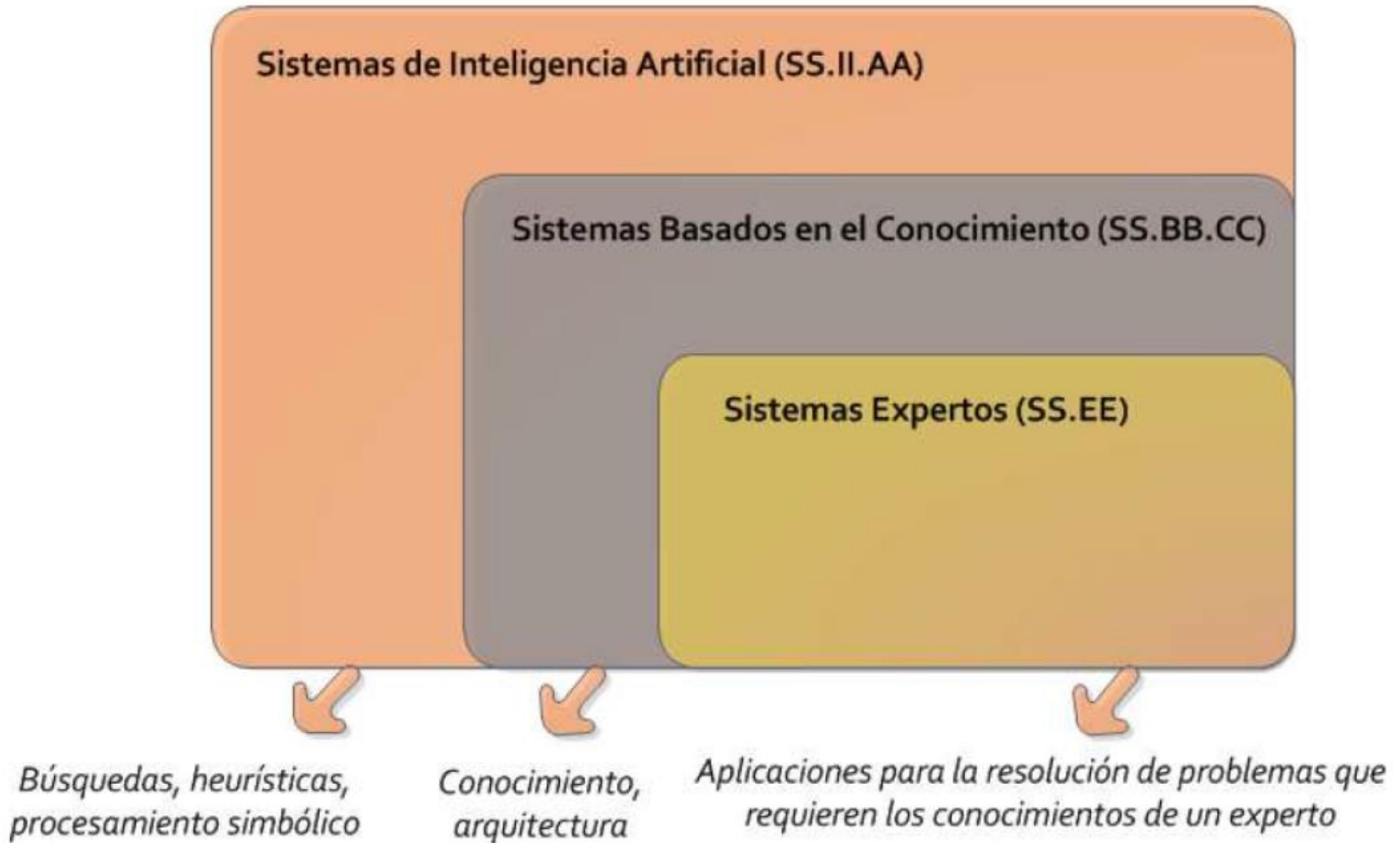
1. Introducción
2. Sistemas basados en el Conocimiento
3. Estructura de un SBC
4. Ingeniería del Conocimiento
  1. IC e ISW
  2. Ciclo de Vida de un SBC
  3. Roles del equipo

## 2. Sistemas Basados en el Conocimiento



- Sistemas Basados en el Conocimiento:
  - Sistema informático que utiliza conocimiento de un dominio de aplicación para resolver problemas de dicho dominio, obteniendo esencialmente las mismas soluciones que obtendría un experto*
  
- Simulan el comportamiento humano de resolución de problemas usando conocimiento para resolver un problema dado representado por hechos específicos
  - Emplean razonamiento simbólico
  - Se denominan **Sistemas Expertos** cuando el *conocimiento que se usa es de humanos (expertos)*
  - Propuestos por Newell y Simon de la Carnegie-Mellon University, a comienzos de los 70.
  - Proporcionan decisiones inteligentes con justificación

## 2. Sistemas Basados en el Conocimiento





## 2. Sistemas Basados en el Conocimiento



- La característica más buscada en un SBC es una alta fiabilidad. El usuario no estará contento si el SBC se equivoca
- La velocidad también importa ya que incluso el diagnóstico más preciso puede ser inútil si llega tarde (caso de una emergencia sanitaria, o una emergencia nuclear)
- Por ello
  - Aplican heurísticas para guiar el razonamiento y reducir el tiempo de búsqueda de una solución
  - Tienen capacidad explicativa, lo que permite al SE revisar su razonamiento y explicar sus decisiones

## 2. Sistemas Basados en el Conocimiento



- ¿Cuándo es adecuado resolver un problema mediante una aproximación basada en el conocimiento (SBC)?
- Cuando se cumple que
  - No hay una solución algorítmica
  - La tarea del dominio la realizan expertos
  - Los expertos no son simples aficionados
  - La tarea no debe ser ni muy difícil ni muy complicada para el experto
  - El problema no debe requerir “sentido común”
  - La utilización del SBC reportará beneficios

## 2. Sistemas Basados en el Conocimiento



- Ventajas:
  - Fácil acceso y disponibilidad de conocimiento experto
  - Coste reducido y Permanencia
  - Fiabilidad y velocidad
  - Capacidad de aprendizaje: entendida como posibilidad de ampliar su base de conocimientos
  - Respuestas no subjetivas
  - Capacidad explicativa: explicación del razonamiento
  - Competitivos con expertos humanos en un dominio pequeño y especializado

## 2. Sistemas Basados en el Conocimiento



- Limitaciones
  - Sentido común: para un Sistema Experto no hay nada obvio.
  - No uso de lenguaje natural
  - No capacidad de aprendizaje
  - Perspectiva global: un experto humano es capaz de distinguir cuáles son las cuestiones relevantes de un problema y separarlas de cuestiones secundarias.
  - Flexibilidad: un humano es sumamente flexible a la hora de aceptar datos para la resolución de un problema.
  - Conocimiento no estructurado: un SBC no es capaz de manejar conocimiento poco estructurado.



1. Introducción
2. Sistemas basados en el Conocimiento
3. Estructura de un SBC
4. Ingeniería del Conocimiento
  1. IC e ISW
  2. Ciclo de Vida de un SBC
  3. Roles del equipo

# 3. Estructura de un SBC



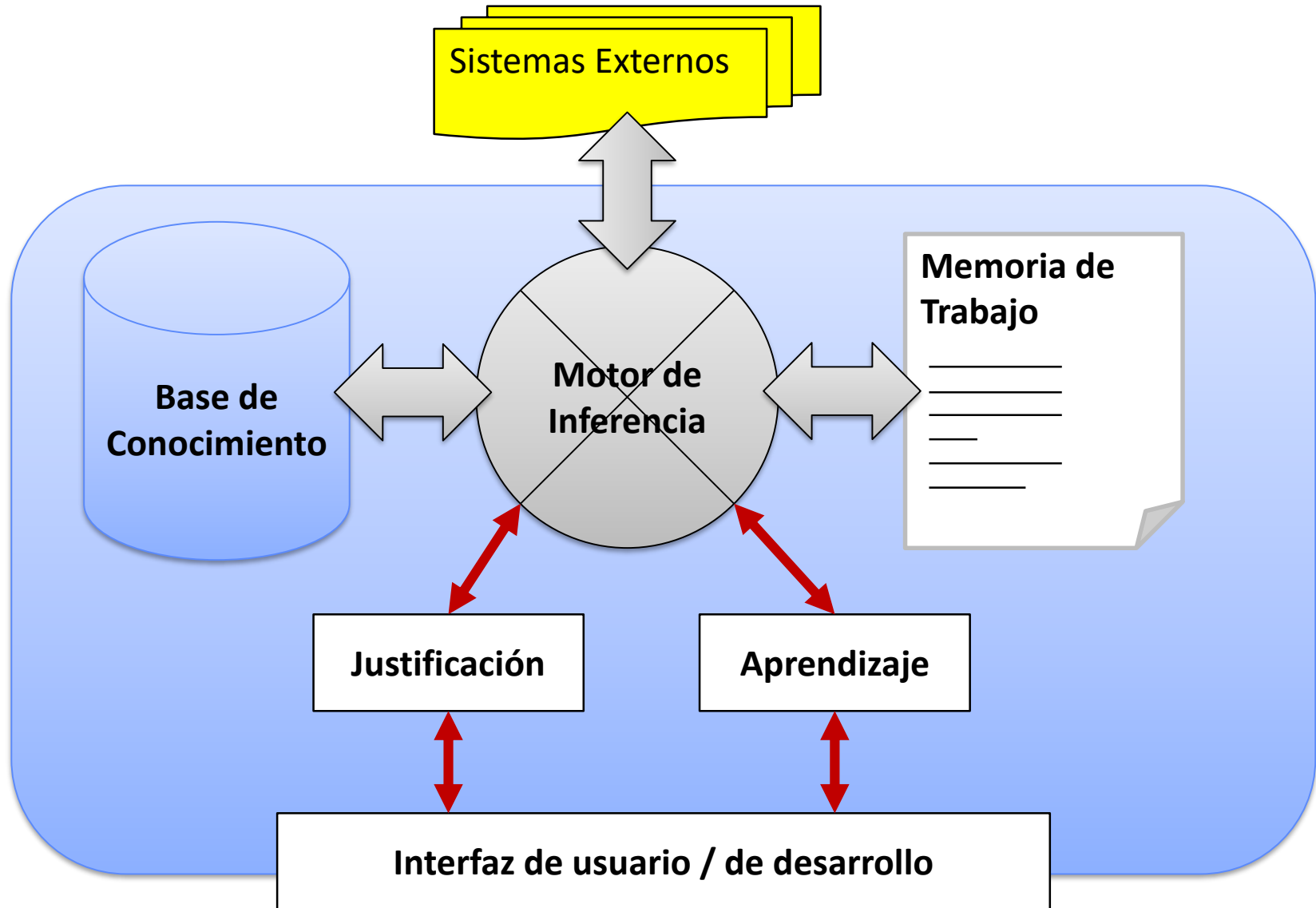
- Componentes principales de un SBC
  - Conocimiento que necesita ser representado en forma de reglas y de hechos
  - Mecanismos que permitan inferir/razonar nuevo conocimiento
  
- Separación entre representación y mecanismo de razonamiento
  - Esto permite usar un mismo mecanismo de razonamiento sobre diversas Bases de Conocimiento
  - Similar a
    - Motor de Bases de Datos y datos almacenados
    - Compilador de un lenguaje y los programas escritos en ese lenguaje
    - Algoritmo de búsqueda y problema representado como un grafo concreto

# 3. Estructura de un SBC



- Estos componentes constituyen *módulos independientes*
  - *Principales*
    - *Base de Conocimiento*
    - *Memoria de Trabajo*
    - *Motor de Inferencia*
  - Otros componentes:
    - Sistema de explicación del resultado (*justificación*)
    - Sistema de adquisición de nuevo conocimiento (*aprendizaje*)
    - Interfaz de usuario / de desarrollo
    - Acceso a sistemas externos (BBDD, otros programas)

### 3. Estructura de un SBC





# 3. Estructura de un SBC



## Base de Conocimiento

*Todo el cuerpo de conocimiento utilizable por el sistema, representado en algún formalismo dado, junto con sus mecanismos de gestión (incorporación, supresión, modificación, consulta, control de consistencia...)*

- El conocimiento del dominio está expresado mediante una ontología
  - Estructura completa de datos que contiene la definición de todas las entidades y sus relaciones dentro del dominio.
  - Es el vocabulario con cuyos términos debe ser descrito todo lo demás
- La base de conocimientos (representación estática) se escribe de acuerdo a la ontología (abstracción) adoptada en el dominio de la aplicación

# 3. Estructura de un SBC



- Sistemas de Producción: Lo usaremos como ejemplo de SBC, pero...

## NO ES el único tipo de SBC

- SBC en el que el conocimiento de resolución de problemas está almacenado habitualmente como reglas de producción (o un formalismo equivalente)
- El conocimiento a alto nivel sobre la resolución del problema está almacenado como **meta-reglas**
  - Permiten dirigir el control de la resolución
    - Activar y desactivar reglas
    - Decidir el orden de ejecución de reglas
    - Decidir estrategias de resolución, tratamiento de excepciones, incertidumbre, ...
  - Son más difíciles de obtener de los expertos

# 3. Estructura de un SBC



- Conocimiento del dominio:
  - Hechos: pieza básica de información (definiciones, tipos, restricciones, valores por defecto)
    - Modelo de hecho: <índice>:<símbolo>(<elemento>\*)
  - Reglas: describen el comportamiento del SBC en función de la información existente
    - Modelo de regla: <Nombre>:SI <Condición>\*  
ENTONCES <Acción>\*
    - Condición:
      - Existencia de cierta información: <patrón>
      - Ausencia de cierta información: -<patrón>
      - Relaciones entre datos
    - Acciones:
      - Incluir nueva información: INCLUIR: <hecho>\*
      - Eliminar información: ELIMINAR: <hecho>\*

# 3. Estructura de un SBC



## Memoria de Trabajo (Pizarra)

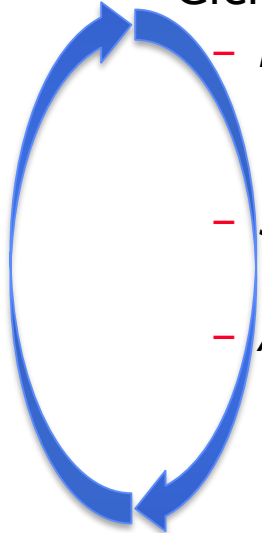
- Guarda los datos iniciales del problema, los hechos obtenidos durante el proceso de razonamiento, hipótesis y decisiones intermedios que el sistema manipula
- Es lo que se sabe del mundo en el momento actual
  - Contiene
    - Agenda: Reglas relevantes para la situación en curso.
    - Solución: Hipótesis candidatas y decisiones generadas hasta el momento
  - Puede guardar otro tipo de información necesaria para el control de la resolución y otros subsistemas
    - Orden de deducción de los hechos
    - Reglas que generaron los hechos
    - Reglas activadas recientemente
    - Puntos de backtracking

# 3. Estructura de un SBC



## Motor de Inferencia (Inference Engine)

- Lleva a cabo el proceso de razonamiento sobre la BC que permite obtener la resolución del problema
  - Ciclo de ejecución para resolver el problema
    - *Detección de reglas aplicables*
      - Son aquellas en las que la *condición* es satisfecha por un hecho
      - Compara las reglas de la BC con los hechos de la Pizarra
    - *Selección de la mejor regla*
      - Estrategia general o guiada por el metaconocimiento
    - *Aplicación de la regla*
      - Cuando la condición de una regla se cumple (*matching*), la regla se *dispara* y se ejecuta la acción asociada (su parte THEN)
      - El disparo puede *asertar* o *retractar* nuevos hechos en la pizarra
  - El proceso sigue hasta alcanzar una conclusión en una cadena de inferencias que se detiene cuando
    - No hay aserción de nuevos hechos o
    - No hay más reglas que disparar



### 3. Estructura de un SBC



- Elementos del Motor de Inferencia
  - *Planificador*: Determina que regla pendiente se ejecutará a continuación. Da a cada elemento de la agenda una prioridad de acuerdo con el plan y las soluciones alcanzadas.
  - *Intérprete*: Ejecuta la regla escogida por el planificador.
  - *Sistema de mantenimiento de la consistencia*: Mantiene una representación consistente de la solución cuando se introducen nuevos hechos
  
- Para optimizar la eficiencia se emplean estrategias más óptimas para representar y manipular las reglas
  - Algoritmo RETE (algoritmo de REdundancia TEmporal)
    - A partir de las reglas se crea inicialmente un grafo (red RETE)
    - Se propaga el contenido de la pizarra inicial a través de la red
    - Cada vez que se produce un cambio en la pizarra (normalmente a través del consecuente de una regla), se propagan los cambios

# 3. Estructura de un SBC



- Resolución de conflictos de reglas:
  - ¿Cómo seleccionar qué regla ejecutar primero cuando hay varias disponibles?
  - Importante por tres motivos:
    - *Por el contenido de la inferencia*: las conclusiones pueden depender del orden en que se apliquen las reglas; las más específicas (y las que tratan las excepciones) deben activarse primero
    - *Por eficiencia*: utilizar la regla adecuada lleva más rápidamente a una conclusión
    - *Por el diálogo que genera*: el orden en que el SBC solicite información debe de ser razonable.

# 3. Estructura de un SBC



- Estrategias de resolución de conflictos de reglas:
  - Mecanismos sencillos
    - Una activación solo se produce una vez
    - Elección aleatoria
  - Uso de estrategias:
    - Control de agenda
      - Tratar la agenda como una pila
      - Tratar la agenda como una cola
      - Mejor regla (pesos)
    - Metarreglas: reglas que razonan sobre otras reglas
    - Regla más específica (numero de condiciones)
    - Activación más reciente (en función de los hechos)
    - Regla menos utilizada



# 3. Estructura de un SBC



## Sistema de Justificación de la solución

- Explica al usuario el porque de las acciones del sistema
  - Debe poder contestar a dos preguntas:
    - ¿Por qué?
    - ¿Cómo?
  - Da credibilidad al sistema
  - Permite detectar deducciones erróneas
  
- Diferentes niveles de justificación:
  - Muestra: Traza de los pasos de resolución
  - Justificación: Razones de los elementos que aparecen en la traza de la resolución (línea de razonamiento, preguntas, hechos, preferencias, subproblemas, ...)

# 3. Estructura de un SBC



## Sistema de Aprendizaje

- El conjunto de problemas que se resuelven está acotado. En algunos dominios es necesario adaptarse al entorno y resolver nuevos problemas
- El aprendizaje puede suceder:
  - Durante el proceso de construcción del SBC: Se substituye o complementa el proceso de adquisición con métodos de aprendizaje inductivo (construir un modelo a partir de ejemplos)
  - Durante el proceso de resolución: Se detectan y corrigen las resoluciones erróneas o se aprenden reglas de control (meta-reglas) que mejoran la eficiencia del proceso de resolución



1. Introducción
2. Sistemas basados en el Conocimiento
3. Estructura de un SBC
4. Ingeniería del Conocimiento
  1. IC e ISW
  2. Ciclo de Vida de un SBC
  3. Roles del equipo

# 4. Ingeniería del Conocimiento



- La expresión **Ingeniería del Conocimiento** se refiere al diseño y construcción metódica de SBC (KBS)
  - *“Adquisición de conocimiento sobre un dominio a partir de fuentes no electrónicas y su conversión a un formato que utilizable por un ordenador para resolver problemas que sólo pueden ser resueltos por personas expertas en el dominio”*
  - *“Engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise”*  
(Feigenbaum & McCorduck, 1983)
  - *“An area of computer science that deals with the development of Expert Systems, integrating specialist human knowledge and expertise into expert computer systems in order to solve complex problems”* (Collins dictionary)

# 4. Ingeniería del Conocimiento



- La Ingeniería del Conocimiento es la parte de “*ingeniería*” de la Inteligencia Artificial.
- Es una disciplina basada en tres pilares:
  - *Herramientas de Inteligencia Artificial*: sistemas de razonamiento artificial, redes bayesianas, razonamiento basado en casos, algoritmos de aprendizaje...
  - *Metodologías propias de la Ingeniería del Conocimiento*: métodos de elicitación (adquisición), sistemas de representación...
  - *Metodologías de Ingeniería del Software*: análisis de requisitos, metodologías de desarrollo, implantación, mantenimiento

## 4.1 IC e ISW



- Al igual que la Ingeniería del SW, la Ingeniería del Conocimiento es una disciplina orientada al diseño, desarrollo y puesta en producción de sistemas informáticos.
- Pero a diferencia de ésta, no es aplicable el ciclo de vida clásico, porque no pueden establecerse unas especificaciones definitivas desde el principio
  - IC presenta similitudes con los desarrollos ágiles y ciclos de vida en espiral, prototipado incremental y metodologías ligeras que se usan en la moderna Ingeniería del SW
  - La ISW las ha heredado en parte de la IC

## 4.1 IC e ISW



- Diferencias entre la IC y la Ingeniería del SW:

Ingeniería del Conocimiento	Ingeniería del SW
Diseña la solución de un problema basándose en conocimientos que un experto tiene en un área específica	El cliente expone requisitos que debe cumplir el sistema
Interacción constante con el experto de principio a fin en el desarrollo del sistema	El usuario normalmente participa solamente en la especificación de requisitos
El conocimiento puede evolucionar durante el uso del sistema o los expertos reconsiderar la forma en la que se ha expresado su conocimiento	Los requisitos se congelan al comienzo del desarrollo y cualquier cambio posterior implica un desarrollo nuevo
La responsabilidad del desarrollo y del mantenimiento recae también en los expertos y en los usuarios finales	Los usuarios se implican en el mantenimiento salvo en el reporte de fallos.

## 4.2 Ciclo de Vida



- Ciclo de Vida de un Sistema Basado en el Conocimiento (Buchanan 1.983)
  - Identificación del problema. Caracterización de los aspectos más importantes del dominio del problema
  - Conceptualización: Adquisición y Validación del conocimiento
    - Se obtiene conocimiento necesarios para resolver el problema de expertos, libros... y se valida
    - Ingeniero del Conocimiento: Persona que traslada el conocimiento de un experto a algún formalismo.
    - Es útil un modelo conceptual (mental) entre el experto y el implementador.
  - Formalización: Representación del conocimiento con las herramientas y esquemas de representación disponibles
    - El conocimiento se representa y codifica en la BC
    - ¿Herramienta de adquisición? ¿Reglas? ¿Redes semánticas? ¿Frames? ¿Esquema de razonamiento? ¿Módulo de explicación?
    - ...

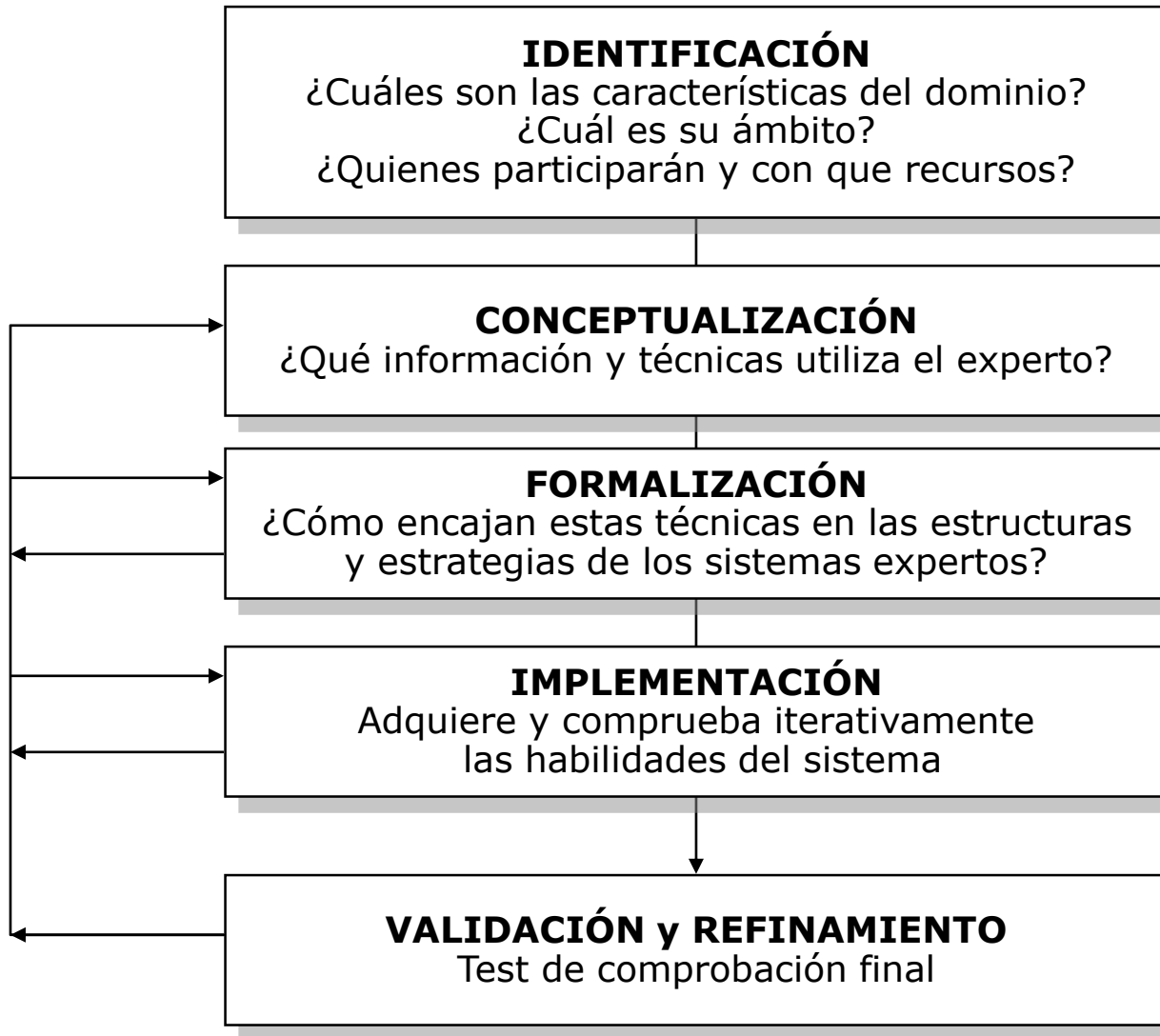


## 4.2 Ciclo de Vida

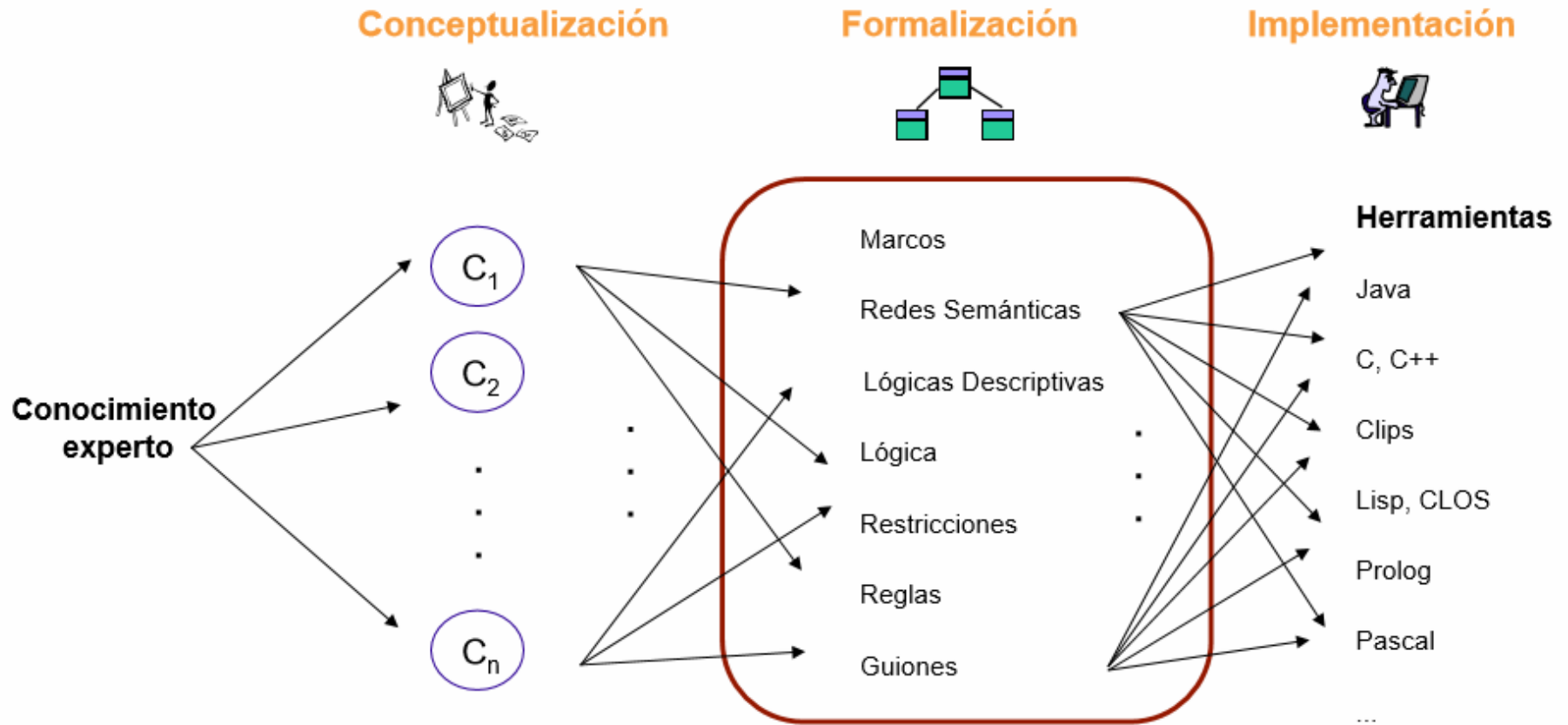


- Implementación de las tareas de conocimiento (inferencia, análisis, diagnóstico, explicación, entrenamiento...)
  - Se dota al sistema de un motor de inferencia para que se puedan alcanzar conclusiones
  - Se capacita al sistema para que pueda explicar cómo llega a las conclusiones que llega
  - Opciones
    - Desarrollo de una herramienta de propósito específico
    - Utilización de algún shell para el desarrollo de SBC's.
- Validación (testeo y evaluación) y refinamiento del SBC
  - El prototipo es comprobado con el experto

## 4.2 Ciclo de Vida



# 4.2 Ciclo de Vida



## 4.2 Ciclo de Vida



- Dos aproximaciones
  - Prototipado:
    - Es el método más popular para el desarrollo de SBC
      - Depende mucho más de la involucración de los usuarios
      - Necesidad de comprobar el comportamiento del sistema según se desarrolla
      - Adquisición del conocimiento y desarrollo del software se pueden combinar en el prototipado
    - Problemas: Desarrollo *ad hoc* e indisciplinado
  - Estructurado:
    - KADS (Knowledge Acquisition and Design process):
      - Metodología de modelado, con una fase rigurosa de análisis antes del diseño
      - El prototipado se utiliza para experimentar, no como metodología

## 4.2 Ciclo de Vida



### ■ Herramientas de desarrollo de SBC's

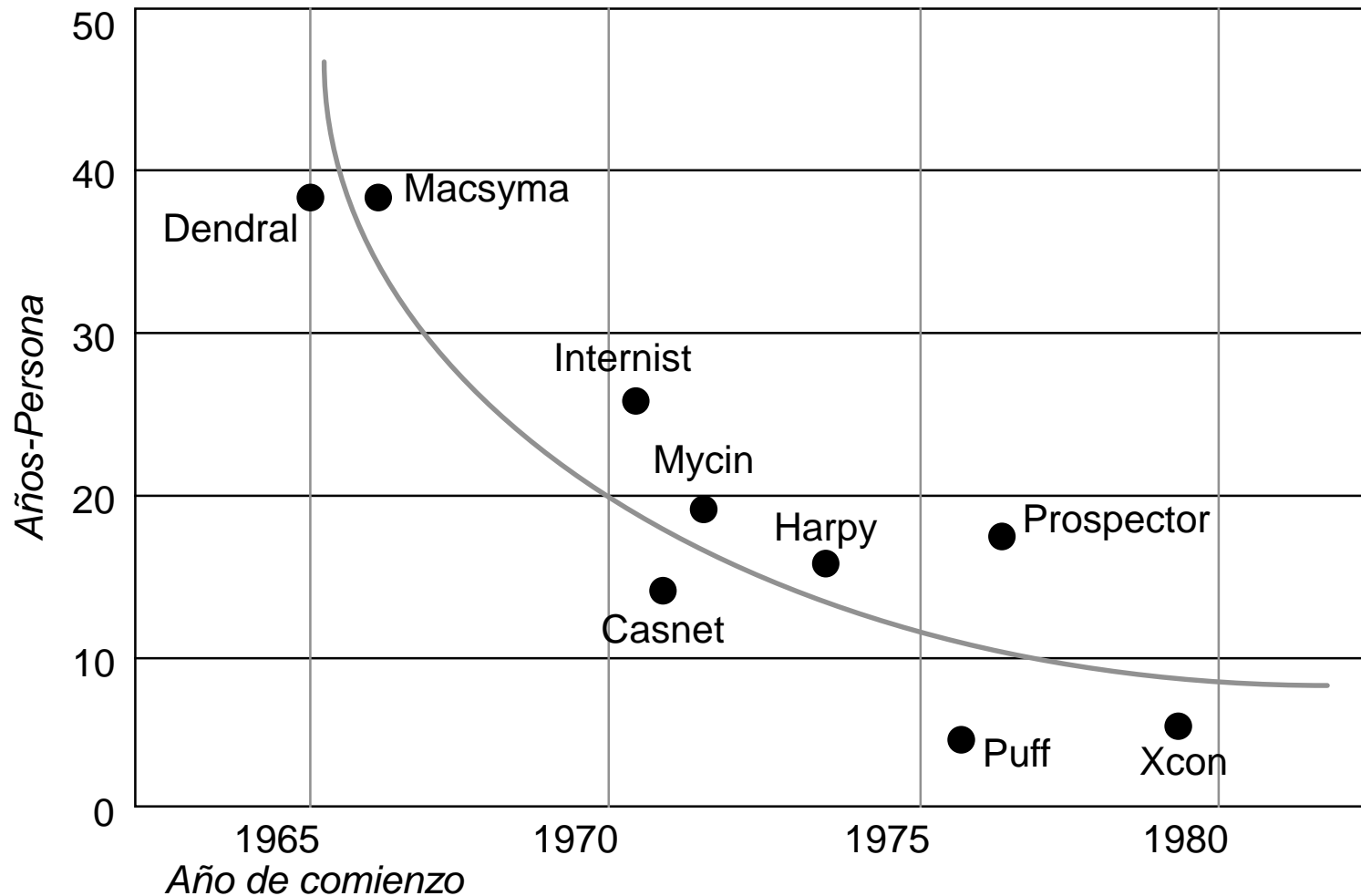
Lisp C Ada	<b>Reglas</b> OPS5 <b>Lógica:</b> PROLOG <b>Objetos:</b> Smalltalk-80 C++ Java	LOOPS <b>CLIPS</b> K-CRAFT ART NEXPERT KAPPA CLOS	KEE VPEXPERT PC+
<i>Lenguajes tradicionales</i>	<i>Paradigmas de programación</i>	<i>Integradores de paradigmas</i>	<i>Shells de desarrollo</i>

- CLIPS: C Language Integrated Production Systems
  - Lenguaje basado en reglas de producción. Relacionado con OPS5
  - Desarrollado en el Johnson Space Center de la NASA
  - Conocimiento: Reglas, objetos y procedimental
  - Implementado en C
  - Software libre

## 4.2 Ciclo de Vida



- Tiempo de desarrollo de un SBC



## 4.3 Roles del equipo



- Hay cinco roles en un equipo de desarrollo de SE:
  - Experto en el dominio
  - Ingeniero del Conocimiento
  - Programador
  - Jefe de proyecto
  - Usuario final
  
- Experto (¡¡¡el rol más importante!!!)
  - Persona con conocimiento y habilidades para resolver problemas en un área específica o dominio
  - Su expertise (conocimiento + experiencia) ha de ser capturada e implementada en el SE
  - El experto debe de
    - Ser capaz de comunicar y transmitir su conocimiento
    - Participar de buena gana en el proceso
    - Comprometer una cantidad importante de tiempo al proyecto

## 4.3 Roles del equipo



- Ingeniero del conocimiento
  - Alguien capaz de diseñar, construir y validar un SE
  - Entrevista al experto para descubrir cómo se soluciona un problema en particular
  - Define los métodos de razonamiento del experto y cómo representarlos en forma de reglas y hechos
  - Elige un SW de desarrollo de SE y valida la implementación final
  
- Programador
  - Responsable de la programación: descripción del conocimiento del dominio en términos entendibles por el ordenador
  - Necesita conocer lenguajes simbólicos y/o shells de desarrollo de SE



## 4.3 Roles del equipo



- Jefe de Proyecto
  - Líder del equipo, responsable de mantener el proyecto en marcha ensamblando las diferentes fases en las que intervienen distintos actores
  - Se debe preocupar de la disponibilidad de todos los roles en todo momento, ya que un proyecto de desarrollo de un SE sigue un ciclo de vida en espiral.
  - Debe conocer técnicas de *rapid development* y *prototipado incremental*
  
- Usuario final
  - Juega un papel en el diseño de la usabilidad del sistema, la interfaz de usuario y la validación post-implantación