- La lógica, proporciona una formulación **simbólica** e **independiente del dominio** de las leyes del pensamiento humano. Este doble carácter hace posible mecanizar sus técnicas y métodos.

- PROBLEMA:
  - Tiene un carácter **estático**, se desarrolló para estudiar objetos matemáticos bien definidos y consistentes, se requieren formas más dinámicas (y menos perfectas) de lógica.
  - Los métodos de la lógica resultan más **caros** en términos computacionales, es necesario reducir sus costes.
- SOLUCIÓN: **Lógica computacional**, modela el conocimiento impreciso, incompleto, dinámico, distribuido. Soporta el razonamiento aproximado, temporal, no monótono,...

Lógicas para Aplicaciones software:
Restricciones: - *Lógica ecuacional*
Extensiones de la lógica: - *Lógica difusa - Lógica Lineal - Lógica Modales*
- Lógica ecuacional:

- Subconjunto de la lógica de primer orden, se eliminan todas las conectivas lógicas. El símbolo de predicado es la (=).
- s=t, dos términos son semánticamente iguales, aunque son diferentes sintácticamente. *Ej: ascii('0') = 48*
- Un conjunto E de ecuaciones con el mismo símbolo "raíz" f en las partes izquierdas de las ecuaciones se describe como "la definición" de f. *even(0)=true - even(X)=even(X-2)*

Lógica para la programación:
Restricciones - Estilo de Lenguaje:
- *Lógica ecuacional - Funcional (Haskell)*
- *Lógica clasual - Relacional (Prolog)*
Extensiones:
- *Lógica many-sorted +tipos*
- *Lógica order-sorted +herencia*
- *Lógica (modal) temporal +concurrencia*
- *Lógica (modal) dinámica +objetos*

- Unificación de lógicas: Lógica de Reescritura (RWL), lógica del cambio que permite especificar la dinámica de un sistema.

- Integración "sin costuras" de distintas características: funciones, y tipos, indeterminismo, concurrencia, reflexión y genericidad.
- Marco unificado en el que pueden definir distintas lógicas: ecuacional, clasual, lineal.
- Existe una lógica temporal, asociada a la RWL, estrictamente más potente que CTL/LTL: la temporal logic of rewriting (LTR).

- Maude:, implementa eficientemente la lógica RWL.

- Soporta de forma natural la especificación formal / modelado / programación en un estilo funcional.
- Distingue entre la parte concurrente y funcional.
- Reescritura módulo listas, conjuntos, multiconjuntos,... mediante atributos ecuacionales.
- Genericidad y tipos ordenados de datos.
- Infraestructura para análisis y verficación formal (alcanzabilidad, model-checking, theorem proving).
- Reflexión como soporte al meta-modelado, ejecución simbólica y construcción rápida de herramientas de soporte.

- Lenguaje Maude

- Sintaxis: Basada en ecuaciones y reglas de reescritura (estilo Haskell, ML, Scheme o Lisp)
- Semántica: Basada en la Lógica de Reescritura (RWL), que modela funciones, concurrencia y objetos.

- Fundamentos de Maude, consta de tres tipos de módulos:

- Módulos funcionales *fmod <conjunto de ecuaciones >endfm*
- Módulos de sistema *mod <conjunto de ecuaciones/reglas de escritura >endm*
- Módulos O2 *omod ... endom*

- E (Ecuaciones):

- Definen funciones confluentes y terminantes.
- Se definen dentro de módulos funcionales.
- E puede incluir un conjunto Ax de Axiomas algebraicos.
- Representan la parte estática del sistema.
- Se aplican de forma determinista.

- R (Reglas de reescritura):

- Definen funciones que pueden ser no confluentes y/o no terminantes.
- Se definen dentro de módulos de sistema.
- Especifican la dinámica del sistema, es decir, acciones que puedan producir transiciones del mismo.

- Paso de reescritura (Maude step), dado un término (o estado) s, un paso de reescritura de t a t' se consigue aplicando una regla de R módulo las ecuaciones de E.
El estado t se simplifica usando las ecuaciones de E hasta alcanzar su forma irreducible (tE) con respecto E.
Una traza de ejecución es una secuencia de Maude steps.