Melhorias ao motor de inferência (PROLOG)

* Generate Metaknowledge
* Cálculo dinâmico do last\_fact e last rule (?)
* Modulo de explicações - “How” e “whyNot”
* Criação de novos factos tendo por base factos criados em runtime.
  + Adição do método “test\_new\_fact” dentro do método “conclude”.
    - Basicamente o test\_new\_fact vai percorrer a lista de regras e ver se deve dar trigger a mais alguma
* Nova função “Evaluate”, que permite avaliar valores de factos criados em runtime.
  + Com o método evaluate\_fact\_values(Fact1, Operator, Fact2) podemos enviar os tipo de factos que estamos à procura e o operador para compararmos os valores.
  + O evaluate/1 compara o valor de uma instância com um valor numérico, exemplo:
    - evaluate(sensor(main, <,10))
      * Obs: neste caso “main” é um identificador único daquele tipo de sensor que está a ler um dado valor
  + O evaluate\_fact\_values/3 (nova função) permite enviar como argumento dois factos e verificar p.e. qual deles tem maior valor, exemplo:
    - evaluate\_fact\_values(power\_output(main,\_),>,power\_input(main,\_))
      * Neste caso, esta função permite ao motor de inferência verificar se o valor do sensor “main” de “power\_output” é maior que o valor do “main” de “power\_input”