

### Hace bien todo

cocinaBien(remy, Plato):-  
plato(Plato).

### es mayor que todos

encargadoDe(PersonaEncargada, Plato, Restaurante):-  
experiencia(Plato, PersonaEncargada, Restaurante, ExperienciaEncargado),  
forall(  
  experiencia(Plato, \_, Restaurante, Experiencia),  
  Experiencia =< ExperienciaEncargado  
).

### entre

satisfaceNecesidad(Integrante, aspiradora(PotenciaRequerida)):-  
tieneHerramienta(Integrante, aspiradora(PotenciaAspiradora)),  
between(0, PotenciaAspiradora, PotenciaRequerida).

### ninguna es

estaDispuestoAHacer(egon, Cliente):-  
not((tareaPedida(Cliente, Tarea, \_),  
  tareaCompleja(Tarea))).

-----  
vocaloidMasFamoso(Vocaloid):-  
  canta(Vocaloid, cancion(\_, \_)),  
  nivelFama(Vocaloid, Nivel),  
  not(  
    (nivelFama(Vocaloid2, Nivel2),  
      Vocaloid \= Vocaloid2,  
      Nivel2 > Nivel)  
  ).

### al menos 2

cantaAlMenos2Canciones(Vocaloid):-  
  canta(Vocaloid, cancion(NombreCancion1,\_)),  
  canta(Vocaloid, cancion(NombreCancion2,\_)),  
  NombreCancion1 \= NombreCancion2.

### recursividad

esConocido(Vocaloid, Conocido):-  
  conoceA(Vocaloid, OtroConocido),  
  esConocido(OtroConocido, Conocido).

### Posiciones

-posicionDeNecesidad(Necesidad, Posicion):-  
  necesidad(Necesidad, Nivel),  
  nivel(Nivel),  
-posicion(Nivel, Posicion).

posicion(Nivel, 1):-  
    not(nivelSuperior(Nivel, \_)).  
-posicion(Nivel, Posicion):-  
    nivelSuperior(Nivel, NivelInferior),  
    posicion(NivelInferior, PosicionInferior),  
    Posicion is PosicionInferior + 1.

#### consecutivos

consecutivos(Mago1, Mago2, Lista):-  
    nth1(IndiceAnterior, Lista, Mago1),  
    nth1(IndiceSiguiente, Lista, Mago2),  
    IndiceSiguiente is IndiceAnterior + 1.