# Lección 1: [Hechos y Reglas](http://pdep-utn.mumuki.io/lessons/1-programacion-logica-hechos-y-reglas)

## Ejercicio 1: Describiendo al mundo

personajeDeFiccion(frodo).  
personajeDeFiccion(harryPotter).

## Ejercicio 2: Hechos

pais(argentina).  
pais(uruguay).  
provincia(santaCruz).  
departamento(canelones).

## Ejercicio 3: Consultas

## Ejercicio 4: Universo cerrado

comeCarne(ramiro).  
comeVerduras(ana).

## Ejercicio 5: Predicados

electrico(pikachu).  
electrico(raichu).  
hierba(bulbasaur).  
fuego(charmander).  
fuego(charizard).

## Ejercicio 6: Aridades

## Ejercicio 7: acontecimiento

acontecimiento(caidaConstantinopla,1453).  
acontecimiento(nacimientoAdaLovelace,1815).  
acontecimiento(inicioSegundaGuerraMundial,1939).  
acontecimiento(fundacionDePixar,1986).

## Ejercicio 8: El barrio de flores

poeta(jorge).  
viveEn(jorge,flores).  
poligrafo(manuel).  
viveEn(manuel,flores).  
tahur(bernardo).  
viveEn(bernardo,flores).  
viveEn(elDiablo,villaCrespo).

## Ejercicio 9: Reglas

## Ejercicio 10: Clausulas

personajeSurrealista(dali).  
personajeSurrealista(Alguien):-apareceEnPinturaSurrealista(Alguien).  
personajeSurrealista(Alguien):-personajeDeCuentoSurrealista(Alguien).

## Ejercicio 11: Reglas más interesantes

## Ejercicio 12: Aristóteles y sus amigos

hombre(hipatia).  
hombre(platon).  
hombre(socrates).  
hombre(aristoteles).  
mortal(Alguien):-hombre(Alguien).

## Ejercicio 13: Los nombres en Flores

leDicen(Nombre,Nombre):-viveEn(Nombre,\_).  
leDicen(Nombre,Apodo):-tahur(Nombre,Apodo).  
leDicen(elDiablo,belcebu).  
leDicen(elDiablo,asmodeo).

# Lección 2: [Práctica Hechos y Reglas](http://pdep-utn.mumuki.io/lessons/2-programacion-logica-practica-hechos-y-reglas)

## Ejercicio 1: losEmpleados

contador(roque).

honesto(roque).

honesto(ana).

joven(roque).

ingeniero(ana).

abogado(cecilia).

## Ejercicio 2: puedeAndar

puedeAndar(contaduria,Empleado):-contador(Empleado),honesto(Empleado).

puedeAndar(ventas,Empleado):-ambicioso(Empleado),tieneExperiencia(Empleado).

puedeAndar(ventas,lucia).

tieneExperiencia(Empleado):-trabajoEn(Empleado,Lugar).

## Ejercicio 3: puedeAndarNuevosSectores

puedeAndar(proyectos,Empleado):-ingeniero(Empleado),conExperiencia(Empleado).  
puedeAndar(proyectos,Empleado):-abogado(Empleado),joven(Empleado).  
puedeAndar(logistica,Empleado):-profesional(Empleado),joven(Empleado).  
puedeAndar(logistica,Empleado):-profesional(Empleado),conExperiencia(Empleado).  
puedeAndar(proyectos,fulgencio).  
puedeAndar(ventas,clotilde).  
profesional(Empleado):-ingeniero(Empleado).  
profesional(Empleado):-abogado(Empleado).

## Ejercicio 4: trabajaEn

empleado(ventas,maria,gerente).  
empleado(ventas,juan,cadete).  
empleado(ventas,roque,cadete).  
empleado(compras,nora,gerente).  
empleado(compras,pedro,cadete).  
empleado(administracion,felipe,gerente).  
empleado(administracion,ana,cadete).  
empleado(administracion,hugo,cadete).  
trabajaEn(Departamento,Empleado):-empleado(Departamento,Empleado,\_).

## Ejercicio 5: esCompatible

esCompatible(Mujer,Hombre):-melancolico(Mujer),sereno(Hombre),mujer(Mujer),hombre(Hombre),Mujer\=Hombre.  
esCompatible(Mujer,Hombre):-decidido(Mujer),reflexivo(Hombre),Mujer\=Hombre,mujer(Mujer),hombre(Hombre).  
esCompatible(Mujer,Hombre):-soniador(Mujer),decidido(Hombre),Mujer\=Hombre,mujer(Mujer),hombre(Hombre).

## Ejercicio 6: esDeseable

esDeseable(Persona):-esCompatible(Persona2,Persona),esCompatible(Persona3,Persona),Persona2\=Persona3.  
esDeseable(Persona):-esCompatible(Persona,Persona2),esCompatible(Persona,Persona3),Persona2\=Persona3.

## Ejercicio 7: gustaDe

gustaDe(luis,nora).  
gustaDe(roque,ana).  
gustaDe(roque,nora).  
gustaDe(marcos,zulema).  
gustaDe(Chicos,zulema):-gustaDe(Chicos,ana).  
gustaDe(juan,Chicas):-gustaDe(roque,Chicas).  
gustaDe(juan,nuria).  
gustaDe(zulema,ana).

## Ejercicio 8: colores

atraeA(mabel,rosa).  
atraeA(ana,rosa).  
atraeA(mara,celeste).  
atraeA(mara,lila).  
atraeA(juan,Color):-colorpastel(Color).  
atraeA(Persona,azul):-mujer(Persona).  
atraeA(pablo,azul).  
atraeA(Persona,rojo):-hombre(Persona),mayor(Persona).  
atraeA(mabel,rojo).  
atraeA(Persona,amarillo):-portenio(Persona).  
atraeA(adrian,amarillo).  
atraeA(ana,naranja).  
atraeA(juan,naranja).  
colorpastel(rosa).  
colorpastel(celeste).  
colorpastel(lila).  
portenio(mabel).  
portenio(mara).  
portenio(pablo).  
mayor(ana).  
mayor(pablo).  
mujer(mabel).  
mujer(ana).  
mujer(mara).  
hombre(juan).  
hombre(pablo).  
hombre(adrian).  
reunion(mabel,viernes).  
reunion(ana,viernes).  
reunion(adrian,viernes).  
reunion(pablo,viernes).  
reunion(mara,sabado).  
reunion(mabel,sabado).  
reunion(adrian,sabado).  
reunion(juan,sabado).  
reunion(juan,domingo).  
reunion(Persona,domingo):-mayor(Persona).  
puedeIluminarseCon(Dia,Color):-atraeA(Persona1,Color),atraeA(Persona2,Color),pareja(Persona1,Persona2),reunion(Persona1,Dia),reunion(Persona2,Dia).  
pareja(Persona1,Persona2):-hombre(Persona1),mujer(Persona2).  
pareja(Persona1,Persona2):-mujer(Persona1),hombre(Persona2).

## Ejercicio 9: Repaso funcional: typeclasses

num(int).  
fractional(float).  
fractional(double).  
num(Tipo):-fractional(Tipo).  
show(int).  
show(float).  
show(double).  
show(bool).  
puedoUsar(sumar,Tipo):-num(Tipo).  
puedoUsar(restar,Tipo):-num(Tipo).  
puedoUsar(multiplicar,Tipo):-num(Tipo).  
puedoUsar(dividir,Tipo):-fractional(Tipo).  
puedoUsar(mostrar,Tipo):-show(Tipo).

# Lección 3: [Negación y cuantificación](http://pdep-utn.mumuki.io/lessons/53-programacion-logica-negacion-y-cuantificacion)

## Ejercicio 1: Conociendo el territorio

viveEn(mariano,avellaneda).  
viveEn(fede,avellaneda).  
viveEn(victoria,versalles).  
viveEn(rodrigo,villaBallester).  
viveEn(tomas,nuniez).  
quedaEn(utn,almagro).  
quedaEn(utn,villaLugano).  
quedaEn(exactas,nuniez).  
quedaEn(river,nuniez).  
quedaEn(racing,avellaneda).

## Ejercicio 2: Existencia, parte 1

tieneAuto(tomas).  
tieneAuto(fede).  
llegaFacil(Persona,\_):-tieneAuto(Persona).  
llegaFacil(batman,\_).

## Ejercicio 3: Existencia, parte 2

## Ejercicio 4: Existencia, parte 3

destinosProximos(Destino1,Destino2):-quedaEn(Destino1,Zona),quedaEn(Destino2,Zona),Destino1\=Destino2.

## Ejercicio 5: ¡Más pruebas!

## Ejercicio 6: ¡Qué bonita vecindad!

sonVecinos(Persona1,Persona2):-viveEn(Persona1,Zona),viveEn(Persona2,Zona),Persona1\=Persona2.

## Ejercicio 7: ¡No, no y no!

## Ejercicio 8: Haciendo dedo

loLleva(Persona1,Persona2):-sonVecinos(Persona1,Persona2),tieneAuto(Persona1),not(tieneAuto(Persona2)).

## Ejercicio 9: Universalidad, parte 1

bienUbicado(Persona):-not((quiereIr(Persona,Lugar),viveEn(Persona,Barrio),quedaEn(Lugar,Barrio2),Barrio\=Barrio2)).

## Ejercicio 10: Universalidad, parte 2

bienUbicado(Persona) :-  
 viveEn(Persona, Zona),   
 forall(quiereIr(Persona, Destino), quedaEn(Destino, Zona)).

## Ejercicio 11: Variables libres

destinoAccesible(Destino):-forall(quiereIr(Persona, Destino),llegaFacil(Persona,Destino)).

## Ejercicio 12: Difícil de estacionar

dificilDeEstacionar(Zona):-viveEn(\_,Zona),forall(viveEn(Persona,Zona),tieneAuto(Persona)).

## Ejercicio 13: ¡Demasiado fácil!, Parte 1

## Ejercicio 14: ¡Demasiado fácil!, parte 2

## Ejercicio 15: Juntando todo

zonaDesierta(Zona):-not(zonaHabitada(Zona)),forall(quedaEn(Lugar,Zona),not(quiereIr(\_,Lugar))).

# Lección 4: [Práctica negación y cuantificación](http://pdep-utn.mumuki.io/lessons/49-programacion-logica-practica-negacion-y-cuantificacion)

## Ejercicio 1: trabajaCon

trabajaCon(Persona1,Persona2):-trabajaEn(Departamento,Persona1),trabajaEn(Departamento,Persona2),Persona1\=Persona2.

## Ejercicio 2: daOrdenes

daOrdenes(Persona1,Persona2):-empleado(Departamento,Persona1,gerente),empleado(Departamento,Persona2,cadete).

## Ejercicio 3: rival

grupoA(colombia).  
grupoA(camerun).  
grupoA(jamaica).  
grupoA(italia).  
grupoB(argentina).  
grupoB(nigeria).  
grupoB(japon).  
grupoB(escocia).  
rival(Pais1,Pais2):-grupoA(Pais1),grupoA(Pais2),Pais1\=Pais2.  
rival(Pais1,Pais2):-grupoB(Pais1),grupoB(Pais2),Pais1\=Pais2.

## Ejercicio 4: compiten

compiten(P1,P2):-gustaDe(P1,P3),gustaDe(P2,P3),P1\=P2.

## Ejercicio 4: fiestaDeCatedra

alumno(luisa,daniel).  
alumno(juan,daniel).  
alumno(diana,nico).  
alumno(nahuel,nico).  
alumno(ana,luisa).  
alumno(claudio,ruben).  
alumno(jose,ruben).  
alumno(alvaro,jose).  
alumno(alvaro,luisa).  
carilindo(brad).  
carilindo(leo).  
carilindo(johnny).  
simpatico(luciano).  
simpatico(lautaro).  
puedeIr(nico).  
puedeIr(daniel).  
puedeIr(Persona):-alumno(Persona,daniel).  
puedeIr(Persona):-alumno(Persona,nico).  
puedeIr(Persona):-carilindo(Persona).  
puedeIr(Persona):-alumno(Persona,Persona2),puedeIr(Persona2).

## Ejercicio 5: Reunión Colorida

atraeA(mabel,rosa).  
atraeA(ana,rosa).  
atraeA(mara,celeste).  
atraeA(mara,lila).  
atraeA(juan,Color):-colorpastel(Color).  
atraeA(Persona,azul):-mujer(Persona).  
atraeA(pablo,azul).  
atraeA(Persona,rojo):-hombre(Persona),mayor(Persona).  
atraeA(mabel,rojo).  
atraeA(Persona,amarillo):-portenio(Persona).  
atraeA(adrian,amarillo).  
atraeA(ana,naranja).  
atraeA(juan,naranja).  
colorpastel(rosa).  
colorpastel(celeste).  
colorpastel(lila).  
portenio(mabel).  
portenio(mara).  
portenio(pablo).  
mayor(ana).  
mayor(pablo).  
mujer(mabel).  
mujer(ana).  
mujer(mara).  
hombre(juan).  
hombre(pablo).  
hombre(adrian).  
reunion(mabel,viernes).  
reunion(ana,viernes).  
reunion(adrian,viernes).  
reunion(pablo,viernes).  
reunion(mara,sabado).  
reunion(mabel,sabado).  
reunion(adrian,sabado).  
reunion(juan,sabado).  
reunion(juan,domingo).  
reunion(Persona,domingo):-mayor(Persona).  
puedeIluminarseCon(Dia,Color):-atraeA(Persona1,Color),atraeA(Persona2,Color),pareja(Persona1,Persona2),reunion(Persona1,Dia),reunion(Persona2,Dia).  
pareja(Persona1,Persona2):-hombre(Persona1),mujer(Persona2).  
pareja(Persona1,Persona2):-mujer(Persona1),hombre(Persona2).

## Ejercicio 6: Repaso funcional: composición

tipo(siguiente,entero,entero).  
tipo(anterior,entero,entero).  
tipo(longitud,lista,entero).  
tipoDeLaComposicion(Funcion2,Funcion1,Entrada,Salida):-tipo(Funcion1,Entrada,Comun),tipo(Funcion2,Comun,Salida).

## Ejercicio 7: TEG

puedenAtacarse(Jugador1,Jugador2):-  
 not(aliados(Jugador1,Jugador2)),  
 ocupa(Pais1,Jugador1,\_),  
 ocupa(Pais2,Jugador2,\_),  
 sonLimitrofes(Pais1,Pais2).  
loLiquidaron(Jugador1):-  
 not(ocupa(\_,Jugador1,\_)).

## Ejercicio 8: TEG 2

ocupaContinente(Jugador,Continente):-  
 forall(estaEn(Continente,Pais),ocupa(Pais,Jugador,\_)).  
   
estaPeleado(Continente):-  
 forall(ocupa(\_,Jugador,\_),(ocupa(Pais,Jugador,\_),estaEn(Continente,Pais))).  
   
seAtrinchero(Jugador):-  
 estaEn(Continente,\_),  
 forall(ocupa(Pais,Jugador,\_),estaEn(Continente,Pais)).

## Ejercicio 9: TEG, La Revancha

elPaisMasFuerte(Pais):-  
 forall(ocupa(Pais,\_,Ejercito),(ocupa(Pais2,\_,Ejercito2),Ejercito>Ejercito2)).

# Lección 5: [Aritmética](http://pdep-utn.mumuki.io/lessons/51-programacion-logica-aritmetica)

## Ejercicio 1: Expresiones, en funcional

## Ejercicio 2: Expresiones, ¿en lógico?

## Ejercicio 3: Consultas vs expresiones

## Ejercicio 4: Individuos vs valores

## Ejercicio 5: ¿1 + 1 = 2?

## Ejercicio 6: Inferencia vs reducción

## Ejercicio 7: Hora de hacer cuentas

## Ejercicio 8: Cuadrado

cuadrado(Numero,Cuadrado):- Cuadrado is Numero\*Numero.

## Ejercicio 9: Máximo

maximo(Num1,Num2,Maximo):- Num1>=Num2, Maximo is Num1.  
maximo(Num1,Num2,Maximo):- Num1<Num2, Maximo is Num2.

# Lección 6: [Inversibilidad](http://pdep-utn.mumuki.io/lessons/34-programacion-logica-inversibilidad)

## Ejercicio 1: Padres e Hijos

padre(homero,bart).  
padre(homero,lisa).  
padre(homero,maggie).  
padre(abraham,homero).  
padre(abraham,herbert).

## Ejercicio 2: Tíos y hermanos

hermano(Hermano1,Hermano2):-padre(Padre,Hermano1),padre(Padre,Hermano2),Hermano1\=Hermano2.  
tio(Tio,Sobrino):-padre(Padre,Sobrino),hermano(Padre,Tio).

## Ejercicio 3: Declaratividad

## Ejercicio 4: Abuelos

abuelo(Abuelo,Nieto):-padre(Padre,Nieto),padre(Abuelo,Padre).  
baboso(Abuelo,Nieto):-abuelo(Abuelo,Nieto),edad(Nieto,Edad),Edad<2.

## Ejercicio 5: Algunas cuentas

cuantaNaftaConsume(Kilometros,Nafta):-Nafta is Kilometros/15.

## Ejercicio 6: ¡Quiero saber más!

## Ejercicio 7: Inversibilidad

## Ejercicio 8: No determinismo, parte 1

## Ejercicio 9: No determinismo, parte 2

## Ejercicio 10: Consultas existenciales

## Ejercicio 11: Problemas de inversibildiad, parte 1

## Ejercicio 12: Problemas de inversibilidad, parte 2

## Ejercicio 13: Generadores

## Ejercicio 14: Mas generadores

buenaNota(Nota):-between(1,10,Nota),Nota>=4.

# Lección 7: [Práctica Inversibilidad](http://pdep-utn.mumuki.io/lessons/3-programacion-logica-practica-inversibilidad)

## Ejercicio 1: area

area(Base,Altura,Area):-Area is Base\*Altura.

## Ejercicio 2: mitad

mitad(Numero,Mitad):-Mitad is Numero/2.

## Ejercicio 3: esAlcoholica

esAlcoholica(Bebida):-esWhisky(Bebida).  
esAlcoholica(Bebida):-esVino(Bebida).

## Ejercicio 4: calcularPrecio

calcularPrecio(Cliente,Bebida,PrecioVenta):-  
 costo(Bebida,Costo),  
 recargo(Cliente,Bebida,Recargo),  
 PrecioVenta is (Costo\*Recargo)/100.  
   
recargo(Cliente,Bebida,Recargo):-  
 esComerciante(Cliente),esAgua(Bebida),Recargo is 125.  
recargo(Cliente,Bebida,Recargo):-  
 esComerciante(Cliente),esGaseosa(Bebida),Recargo is 100.  
recargo(Cliente,Bebida,Recargo):-  
 esComerciante(Cliente),esAlcoholica(Bebida),Recargo is 130.  
recargo(Cliente,Bebida,Recargo):-  
 esParticular(Cliente),esAgua(Bebida),Recargo is 130.  
recargo(Cliente,Bebida,Recargo):-  
 esParticular(Cliente),esGaseosa(Bebida),Recargo is 140.  
recargo(Cliente,Bebida,Recargo):-  
 esParticular(Cliente),esAlcoholica(Bebida),esNacional(Bebida),Recargo is 160.  
recargo(Cliente,Bebida,Recargo):-  
 esParticular(Cliente),esAlcoholica(Bebida),esImportada(Bebida),Recargo is 170.

## Ejercicio 5: triple

triple(Numero,Triple):-Triple is Numero\*3.

## Ejercicio 6: Repaso funcional: Aplicación Parcial

tipoDeAplicacion(Aridad,Argumentos,parcial):-  
 AridadMenosUno is Aridad-1,  
 between(1,AridadMenosUno,Argumentos).  
tipoDeAplicacion(Aridad,Aridad,total).  
tipoDeAplicacion(\_,0,sinAplicar).

## Ejercicio 7: puntajeLanzamientoPrecision

puntajeLanzamientoPrecision(Persona,Puntaje):-  
 metrosLanzamiento(Persona,Metros),calcPuntaje(Metros,Puntaje).  
   
calcPuntaje(Metros,10):-  
 between(9,11,Metros).  
calcPuntaje(Metros,6):-  
 between(7,8,Metros).  
calcPuntaje(Metros,6):-  
 between(12,15,Metros).  
calcPuntaje(Metros,0):-  
 Metros<7.  
calcPuntaje(Metros,0):-  
 Metros>16.

## Ejercicio 8: puntajeFuerzaMartillo

puntajeFuerzaMartillo(Persona,Puntaje):-  
 metrosMartillo(Persona,Metros),  
 calcPuntaje(Metros,Puntaje).  
  
calcPuntaje(Metros,3):-  
 Metros=<4.  
calcPuntaje(Metros,6):-  
 between(5,10,Metros).  
calcPuntaje(Metros,Puntaje):-  
 Metros>10,Puntaje is 9+ ((Metros-10) \* 2).

## Ejercicio 9: puntajeEquilibrioEscoba

puntajeEquilibrioEscoba(Persona,Puntaje):-  
 metrosEscoba(Persona,Metros),calcPuntaje(Metros,Puntaje).  
  
calcPuntaje(Metros,Puntaje):-  
 Puntaje is round(Metros/3).

## Ejercicio 10: puntajeTotal

puntajeTotal(Persona,0):-  
puntajeLanzamientoPrecision(Persona,Precision),  
puntajeFuerzaMartillo(Persona,Martillo),  
puntajeEquilibrioEscoba(Persona,Escoba),  
algunoMenorCinco(Precision,Martillo,Escoba).  
  
algunoMenorCinco(Precision,\_,\_):-Precision<5.  
algunoMenorCinco(\_,Martillo,\_):-Martillo<5.  
algunoMenorCinco(\_,\_,Escoba):-Escoba<5.  
  
puntajeTotal(Persona,Total):-  
puntajeLanzamientoPrecision(Persona,Precision),  
puntajeFuerzaMartillo(Persona,Martillo),  
puntajeEquilibrioEscoba(Persona,Escoba),  
Total is Precision+Martillo+Escoba.

## Ejercicio 11: asesinato

asesino(Victima,Persona):-odiaA(Persona,Victima),viveEnMansion(Persona),  
 not(esMasRicoQueAgatha(Persona)).  
viveEnMansion(agatha).  
viveEnMansion(charles).  
viveEnMansion(carnicero).  
odiaA(charles,Persona):-  
 viveEnMansion(Persona),  
 not(odiaA(agatha,Persona)).  
odiaA(agatha,Persona):-  
 viveEnMansion(Persona),  
 Persona\=carnicero.  
esMasRicoQueAgatha(Persona):-  
 viveEnMansion(Persona),  
 not(odiaA(carnicero,Persona)).  
odiaA(carnicero,Persona):-odiaA(agatha,Persona).

## Ejercicio 12: Equipo al horno

alHorno(Equipo):-  
 perteneceA(\_,Equipo),  
 forall(perteneceA(Persona,Equipo),debilOFueraDeJuego(Persona)).  
   
debilOFueraDeJuego(Persona):-  
 fueraDeJuego(Persona).  
debilOFueraDeJuego(Persona):-  
 nivelDeEnergia(Persona,Nivel),Nivel<10.

## Ejercicio 13: Equipo cipayo

equipoCipayo(Equipo1):-  
 perteneceA(Part1,Equipo1),  
 perteneceA(Part2,Equipo1),  
 perteneceA(Part1,Equipo2),  
 perteneceA(Part2,Equipo3),  
 Part1\=Part2,  
 Equipo1\=Equipo2,  
 Equipo1\=Equipo3.

## Ejercicio 14: Equipo Abandonó

abandono(Equipo):-equipo(Equipo),not(perteneceA(\_,Equipo)).

# Lección 8: [Functores](http://pdep-utn.mumuki.io/lessons/4-programacion-logica-functores)

## Ejercicio 1: Tuplas: agrupando indiviuos

## Ejercicio 2: Más Tuplas

personaje(arya, (14, mujer)).  
personaje(cersei, (34 , mujer)).

## Ejercicio 3: Functores

personaje(arya,stark(14,mujer)).  
personaje(cersei,lannister(34,mujer)).

## Ejercicio 4: Patrones de Functores

esStarkAdulto(Nombre):-personaje(Nombre,stark(Edad,\_)),Edad>15.

## Ejercicio 5: Patrones de Functores 2

personajeAdulto(Nombre):-personaje(Nombre,stark(Edad,\_)),Edad>15.  
personajeAdulto(Nombre):-personaje(Nombre,lannister(Edad,\_)),Edad>15.

## Ejercicio 6: Diferentes pero no tanto

personajeAdulto(Nombre):-personaje(Nombre,Personaje),esAdulto(Personaje).  
personajeAdulto(Nombre):-personaje(Nombre,Personaje),esAdulto(Personaje).  
esAdulto(lannister(Edad,mujer)):-Edad>15.  
esAdulto(stark(Edad,mujer)):-Edad>15.

## Ejercicio 7: Functores vs predicados

esPeligroso(Nombre):-esPersonajePeligroso(Nombre),personaje(Nombre,\_).  
  
esPersonajePeligroso(Nombre):-personaje(Nombre,stark(\_,\_)).  
esPersonajePeligroso(Nombre):-personaje(Nombre,lannister(Oro)),Oro>=300.

## Ejercicio 8: Diferentes aridades

esPersonajePeligroso(nightwatch(\_, lobo(\_))).

## Ejercicio 9: Algo llamado polimorfismo

cuantoSabePersonaje(lannister(\_), mucho).  
cuantoSabePersonaje(nightwatch(\_), poco).  
cuantoSabePersonaje(stark(\_, \_), poco).  
cuantoSabePersonaje(nightwatch(\_, lobo(\_)), nada).

## Ejercicio 10: Algo llamado polimorfismo 2

cuantoSabe(Nombre,Cuanto):-personaje(Nombre, Personaje),cuantoSabePersonaje(Personaje,Cuanto).

# Lección 9: [Práctica Functores](http://pdep-utn.mumuki.io/lessons/5-programacion-logica-practica-functores)

## Ejercicio 1: servidores

requiereAtencion(Servidor,Evento,Prioridad):-servidor(Servidor,\_,\_),eventosServidor(Evento,Servidor),prioridad(Evento,Prioridad).  
  
eventosServidor(rebooteo(ServidorInicial),ServidorFinal):-afectados(ServidorInicial,ServidorFinal).  
eventosServidor(cuelgue(Servidor),Servidor).  
eventosServidor(corteDeLuz(Fila),Servidor):-servidor(Servidor,Fila,\_).  
prioridad(corteDeLuz(\_),inmediata).  
prioridad(rebooteo(\_),normal).  
prioridad(cuelgue(Servidor),Prioridad):-servidor(Servidor,\_,Criticidad),criticidad(Prioridad,Criticidad).  
criticidad(inmediata,Criticidad):- Criticidad is 1 ;Criticidad is 2.  
criticidad(normal,Criticidad):- Criticidad is 3 ;Criticidad is 4.  
  
afectados(ServidorInicial,ServidorFinal):-esCliente(ServidorFinal,ServidorInicial).  
afectados(ServidorInicial,ServidorInicial).

## Ejercicio 2: Mes complicado

complicado(Mes):-  
 ocurrio(\_,Mes),  
 forall(servidor(Servidor,\_,\_),(ocurrio(Evento,Mes),requiereAtencion(Servidor,Evento,\_))).

## Ejercicio 3: agenciaMatrimonial2

## Ejercicio 4: busquedaDelTesoro

destinoPosible(Persona,Ciudad):-  
 nivelActual(Persona,Nivel),  
 tarea(Nivel,buscar(\_,Ciudad)).  
   
idiomaUtil(Persona,Idioma):-  
 destinoPosible(Persona,Ciudad),  
 idioma(Ciudad,Idioma).  
   
excelenteCompaniero(Primero,Segundo):-  
 nivelActual(Primero,\_),nivelActual(Segundo,\_),Primero\=Segundo,  
 forall(idiomaUtil(Primero,Idioma),habla(Segundo,Idioma)).  
  
interesante(Nivel):-tarea(Nivel,buscar(\_,Ciudad)),idioma(Ciudad,italiano).  
interesante(Nivel):-nivelActual(\_,Nivel),forall(tarea(Nivel,buscar(Cosa,\_)),esViva(Cosa)).  
esViva(arbol).  
esViva(perro).  
esViva(flor).  
  
complicado(Persona):-idiomaUtil(Persona,Idioma),not(habla(Persona,Idioma)).  
complicado(Persona):-not(nivelActual(Persona,basico)),capital(Persona,Capital),Capital<1500.  
complicado(Persona):-nivelActual(Persona,basico),capital(Persona,Capital),Capital<500.  
homogeneo2(Nivel) :- nivelActual(\_, Nivel), forall(tarea(Nivel, buscar(Cosa, \_)), not((tarea(Nivel, buscar(Cosa2, \_)), Cosa2 \= Cosa))).  
homogeneo(Nivel):-  
 nivelActual(\_,Nivel),  
 tarea(Nivel,buscar(Cosa1,\_)),  
 forall(tarea(Nivel,buscar(Cosa2,\_)),  
 Cosa1=Cosa2).

## Ejercicio 5: tareas

anterior(Tarea1,Tarea3):-precede(Tarea1,Tarea2),anterior(Tarea2,Tarea3).  
anterior(Tarea1,Tarea2):-precede(Tarea1,Tarea2).  
  
simple(Tarea):-tarea(Tarea,Horas,Quien),simple2(Horas,Quien).  
simple2(\_,programador).  
simple2(\_,tester).  
simple2(Tiempo,\_):-Tiempo<40.  
  
riesgo(Tarea):-not(realizada(Tarea)),cuarentaOMas(Tarea).  
cuarentaOMas(Tarea):-tarea(Tarea,Tiempo,\_),Tiempo>=40.  
  
meFaltaPara(Tarea,Faltantes):-anterior(Faltantes,Tarea),not(realizada(Faltantes)).  
  
puedoHacer(Tarea):-  
 tarea(Tarea,\_,\_),  
 not(realizada(Tarea)),  
 forall(  
 meFaltaPara(Tarea,Precedentes),  
 realizada(Precedentes)  
 ).

## Ejercicio 6: tareasMultiples

## Ejercicio 7: domino

# Lección 10: [Más allá del paradigma: findall](http://pdep-utn.mumuki.io/lessons/35-programacion-logica-mas-alla-del-paradigma-findall)

## Ejercicio 1: El orden no importa... ¿verdad?

puedePrestar(Alguien, Otro, Objeto) :-  
 tiene(Alguien, Objeto),  
 necesita(Otro, Objeto),  
 Alguien \= Otro.

## Ejercicio 2: El orden de las cláusulas

## Ejercicio 3: Orden de las soluciones

## Ejercicio 4: Findall

empleadosNecesitados(Lista):-findall(Necesitado, necesita(Necesitado, \_),Lista).

## Ejercicio 5: De muchas soluciones a una

herramientasDemandadas(Lista):-findall(Herramienta, necesita(\_,Herramienta), Lista).

## Ejercicio 6: cuantoFalta

cuantoFalta(Cantidad):-findall(Herramienta,necesita(\_,Herramienta),Herramientas),list\_to\_set(Herramientas,SinRepetidos),length(SinRepetidos,Cantidad).

## Ejercicio 7: Usar findall y no morir en el intento

costoTotalDeHerramientasDemandadas(CostoTotal):-findall(Costo,(herramientaDemandada(Herramienta),costoHerramienta(Herramienta,Costo)),LstCosto),sum\_list(LstCosto,CostoTotal).

## Ejercicio 8: Findall y la inversibilidad

herramientasDemandadasPor(Persona,LstHerramientas):-necesita(Persona,\_),findall(Herramienta,necesita(Persona,Herramienta),LstHerramientas).

## Ejercicio 9: Findall en todos lados

noLaNecesitaNadie(Herramienta):-  
 not(necesita(Persona,Herramienta)).

# Lección 11: [Práctica Listas y Findall](http://pdep-utn.mumuki.io/lessons/6-programacion-logica-practica-listas-y-findall)

## Ejercicio 1: puntajeCompetidor

buenPasar(Persona):-ingreso(Persona,Mes,Gano),Mes=enero,Gano>2300.  
buenPasar(Persona):-ingreso(Persona,Mes,Gano),Mes\=enero,Gano>3000.  
buenPasar(Persona):-ingreso(Persona,Mes1,Gano1),Gano1>2500,ingreso(Persona,Mes2,Gano2),Gano2>2500,Mes1\=Mes2.  
mesFilial(Persona,Mes):-padre(Hijo,Persona),ingreso(Persona,Mes,Gano),ingreso(Hijo,Mes,Gano2),Gano2>Gano.  
ingresoTotal(Persona,Ingreso):-findall(Gano,ingreso(Persona,Mes,Gano),Ingresos), sumlist(Ingresos,Ingreso).  
ingresoFamiliar(Persona,IngFamiliar):-ingresoTotal(Persona,Ingreso),findall(Gano,(padre(Hijo,Persona),ingreso(Hijo,Mes,Gano)),IngresosHijosLST),sumlist(IngresosHijosLST,IngresoCasi),IngFamiliar is Ingreso+IngresoCasi.

## Ejercicio 2: competidorDescalificado

competidorDescalificado(Competidor):-puntajes(Competidor,LstSaltos),length(LstSaltos,Cantidad),Cantidad>=6.

## Ejercicio 3: competidorClasifica

competidorClasifica(Competidor):-puntajes(Competidor,LstPuntajes),clasifica(LstPuntajes).  
  
clasifica(LstPuntajes):-sum\_list(LstPuntajes,Suma),Suma>=28  
 ;nth1(Pos1,LstPuntajes,Salto1),nth1(Pos2,LstPuntajes,Salto2),Salto1>=8,Salto2>=8,Pos1\=Pos2.

## Ejercicio 4: buenPasar

buenPasar(Persona):-ingreso(Persona,enero,Dinero),Dinero>2200.  
buenPasar(Persona):-ingreso(Persona,\_,Dinero),Dinero>3000.  
buenPasar(Persona):-ingreso(Persona,Mes1,Dinero1),ingreso(Persona,Mes2,Dinero2),Dinero1>2500,Dinero2>2500,Mes1\=Mes2.

## Ejercicio 5: mesFilial

mesFilial(Persona,Mes):-padre(Hijo,Persona),ingreso(Persona,Mes,Dinero1),ingreso(Hijo,Mes,Dinero2),Dinero2>Dinero1.

## Ejercicio 6: ingresoTotal

ingresoTotal(Persona,IngresoTotal):-  
 ingreso(Persona,\_,\_),  
 findall(Dinero,ingreso(Persona,\_,Dinero),LstDinero),  
 sum\_list(LstDinero,IngresoTotal).

## Ejercicio 7: ingresoFamiliar