

# iLógico

Los siguientes fragmentos de código andan, su lógica es correcta y no poseen errores de sintaxis **pero**: todas las soluciones poseen al menos un error conceptual.

Para cada punto:

- Analizar su lógica.
- Dar la cantidad de cláusulas, predicados y su aridad.
- Identificar los errores conceptuales, si los hubiera
- Proponer una solución que aproveche los conceptos del paradigma.
- Analizar si los predicados son totalmente inversibles e *inversibilizarlos* si no lo fueran.

## Punto 1

```
todosSiguenA(Rey) :-  
    personaje(Rey),  
    not((personaje(Personaje), not(sigueA(Personaje, Rey))))).
```

```
sigueA(Alguien, Alguien).  
sigueA(lyanna, jon).  
sigueA(jorah, daenerys).  
%% etc
```

## Punto 2

```
baresCopados(Ciudad, Bares) :-  
    findall(Bar, (puntoDeInteres(bar(CantVarCer), Ciudad), CantVarCer > 4), Bares).
```

```
museosCopados(Ciudad, Museos) :-  
    findall(Museo, puntoDeInteres(museo(cienciasNaturales), Ciudad), Museos).
```

```
ciudadInteresante(Ciudad) :-  
    antigua(Ciudad),  
    baresCopados(Ciudad, Bares),  
    museosCopados(Ciudad, Museos),  
    length(Bares, CantidadBares),  
    length(Museos, CantidadMuseos),  
    CantidadLugaresCopados is CantidadBares + CantidadMuseos,  
    CantidadLugaresCopados > 10.
```

## Punto 3

```
inFraganti(Delito, Delincuente) :-  
    cometio(Delito, Delincuente),  
    findall(Testigo, testigo(Delito, Testigo), Testigos),  
    length(Testigos, Cantidad),  
    Cantidad > 0.
```

#### **Punto 4**

```
viejoMaestro(Pensador) :-  
    forall(pensamiento(Pensador, Pensamiento), llegaANuestrosDias(Pensamiento)).
```

#### **Punto 5**

```
numeroDeLaSuerte(Persona, Numero) :-  
    diaDelNacimiento(Persona, Numero).
```

```
numeroDeLaSuerte(joaquin, Numero) :-  
    Numero is 8.
```

#### **Punto 6**

```
obraMaestra(Compositor, Obra) :-  
    compositor(Compositor, Obra),  
    forall(movimiento(Obra, Movimiento), cumpleCondiciones(Movimiento)).
```

#### **Punto 7**

```
puedeComer(analia, Comida) :-  
    ingrediente(Comida, _),  
    forall(ingrediente(Comida, Ingrediente),  
        (not(contieneCarne(Ingrediente),  
            not(contieneHuevo(Ingrediente),  
            not(contieneLeche(Ingrediente))))).
```

```
puedeComer(evaristo, asado).  
%% etc
```

#### **Punto 8**

```
costoEnvio(Paquete, PrecioTotal) :-  
    findall(PrecioItem, precioItemPaquete(Paquete, PrecioItem), Precios),  
    sumlist(Precios, PrecioTotal).
```

```
precioItemPaquete(Paquete, Precio) :-  
    itemPaquete(Paquete, libro(Precio)).
```

```
precioItemPaquete(Paquete, Precio) :-  
    itemPaquete(Paquete, mp3(_, Duracion)),  
    Precio is Duracion * 0.42.
```

```
precioItemPaquete(Paquete, PrecioOferta) :-  
    itemPaquete(Paquete, productoEnOferta(_, PrecioOferta)).
```