

```
amaALosGatos(X) :-  
tieneMascota(X, Y), gato(Y).
```

La cabeza unifica con el primer literal de la consulta. Hallamos la sustitución  $X := Z$ , sustituimos y reemplazamos por el cuerpo de la cláusula.

1. `amaALosGatos(Z).`

$\{X := Z\}$

```
tieneMascota(john, garfield).
```

Cuando la cláusula es un hecho, Prolog simplemente se fija que el literal unifique con el hecho.

2. `tieneMascota(Z, Y),  
gato(Y).`

$\{Z := \text{john},$   
 $Y := \text{garfield}\}$

```
tieneMascota(john, odie).
```

$\{Z := \text{john},$   
 $Y := \text{odie}\}$

```
gato(garfield).
```

3. `gato(garfield).`

4. ☐

El motor de Prolog encontró la solución y muestra por consola "Z = john". Si apretamos ;, podemos pedir más soluciones, y Prolog intentará resolver `tieneMascota(Z, Y)` a partir de `tieneMascota(john, odie)`.

5. `gato(odie).` No hay ninguna cláusula del programa que unifique con esta. Entonces, no hay más soluciones, y Prolog muestra por consola "false".



### Programa

```
gato(garfield).  
tieneMascota(john, garfield).  
tieneMascota(john, odie).  
  
amaALosGatos(X) :-  
tieneMascota(X, Y) , gato(Y).
```