

```
(defrule tipo-trianguulo
(initial-fact)
=>
(printout t "*****Tipo de Triangulo*****" crlf)
(printout t " 1) Triangulos por sus lados" crlf)
(printout t " 2) Triangulos por sus angulos" crlf)
(assert (opcion(read)))
)
```

```
(defrule triangulo-lados
(opcion 1)
=>
(printout t "*****Escoge una opcion*****" crlf)
(printout t " 1) El triangulo tiene sus 3 lados iguales" crlf)
(printout t " 2) El triangulo tiene sus 2 lados iguales" crlf)
(printout t " 3) El triangulo no tiene lados iguales" crlf)
(assert (lados(read)))
)
```

```
(defrule triangulo-equilatero
(lados 1)
=>
(printout t "Es un triangulo Equilatero" crlf)
)
```

```
(defrule triangulo-isosceles
(lados 2)
=>
(printout t "Es un triangulo Isosceles" crlf)
)
```

```
(defrule triangulo-escaleno
(lados 3)
=>
(printout t "Es un triangulo Escaleno" crlf)
)
```

```
(defrule triangulo-angulos
(opcion 2)
=>
(printout t "*****Escoge una opcion*****" crlf)
(printout t " 1) El triangulo tiene un angulo de 90° " crlf)
(printout t " 2) El triangulo tiene un angulo menor a 90° " crlf)
(printout t " 3) El triangulo tiene un angulo mayor a 90° " crlf)
(assert (angulos(read)))
)
```

```
(defrule triangulo-retangulo
(angulos 1)
=>
(printout t "Es un triangulo Rectangulo" crlf)
)
```

```
(defrule triangulo-acutangulo
(angulos 2)
=>
(printout t "Es un triangulo Acutangulo" crlf)
)
```

```
(defrule triangulo-obtusangulo
(angulos 3)
=>
(printout t "Es un triangulo Obtusangulo" crlf)
)
```