

## Parcial II

*Luis Felipe Garzón Bonilla 1089931169*

*Steven Grisales López 1034289634*

*Santiago Aristizabal Sepúlveda 1089932388*

Se solicita crear un programa en Prolog que modele el triángulo, el cuadrado y el rectángulo. Se deben utilizar estructuras y listas para dicho modelamiento.

### Panel de Hechos.

```
% Definición de un segmento
segmento(punto(X1,Y1),punto(X2,Y2)).

% Regla para determinar si un segmento es horizontal
horizontal(segmento(punto(X1, Y), punto(X2, Y))) :-
    X1 \= X2, !.

% Regla para determinar si un segmento es vertical
vertical(segmento(punto(X, Y1), punto(X, Y2))) :-
    Y1 \= Y2, !.

% Regla para determinar si un conjunto de segmentos forman un polígono cerrado
poligono_cerrado(Segmentos) :-
    last(Segmentos, segmento(punto(_,Y), punto(X1,Y))),
    nth0(0, Segmentos, segmento(punto(X2,Y), _)),
    X1 =:= X2.
```

En este panel se declaran las reglas y hechos necesarios para las próximas verificaciones de triángulo, rectángulo y cuadrado. En primera instancia, se declara que un segmento está formado por dos puntos de diferentes coordenadas X y Y.

Para verificar si un segmento es horizontal, vertical Se tienen que comparar las coordenadas de los puntos, siendo para horizontal, que la coordenada x debe ser igual en ambos puntos, pero la coordenada Y debe ser diferente. Este caso viceversa para la comprobación vertical.

Estas dos reglas fueron necesarias para el apartado de rectángulo, no hicieron falta a la hora de verificar un cuadrado o triángulo, pero la de polígono cerrado fue necesaria en las 3 ocasiones.

## Reglas de figuras geométricas.

### ➤ Triángulo

*% Regla para determinar si un conjunto de segmentos es un triángulo*

```
triangulo(Segmentos) :-  
    length(Segmentos, 3),  
    poligono_cerrado(Segmentos).
```

Para determinar si es un triángulo, se verifica que existan 3 segmentos y se verifica que el polígono sea cerrado, lo que nos da como resultado un triángulo.

```
⚙ triangulo([segmento(punto(1,1), punto(2,2)), segmento(punto(2,2), punto(3,1)), segmento(punto(3,1), punto(1,1))]).  
true
```

### ➤ Cuadrado

*% Regla para determinar si un conjunto de segmentos es un cuadrado*

```
cuadrado(Segmentos) :-  
    length(Segmentos, 4),  
    poligono_cerrado(Segmentos),  
    member(segmento(punto(X,Y), punto(X1,Y)), Segmentos),  
    member(segmento(punto(X1,Y), punto(X1,Y1)), Segmentos),  
    member(segmento(punto(X1,Y1), punto(X,Y1)), Segmentos),  
    member(segmento(punto(X,Y1), punto(X,Y)), Segmentos), !.
```

Para determinar si es un cuadrado, se verifica que existan 4 segmentos, se verifica el polígono sea cerrado y rectificamos que hayan cuatro puntos unidos entre sí, por ejemplo, el segmento 1 se conforma de los puntos (X,Y) y (X1,Y), para después ir verificando los otros 3 segmentos que se van conectando con el segundo punto del segmento anterior, dándonos así como resultado un cuadrado.

```
⚙ cuadrado([segmento(punto(1,1), punto(1,2)), segmento(punto(1,2), punto(2,2)), segmento(punto(2,2), punto(2,1)),  
segmento(punto(2,1), punto(1,1))]).  
true
```

### ➤ Rectángulo

*% Regla para determinar si un conjunto de segmentos es un rectángulo*

```
rectangulo(Segmentos) :-
    length(Segmentos, 4),
    poligono_cerrado(Segmentos),
    member(segmento(punto(X,Y), punto(X1,Y)), Segmentos),
    member(segmento(punto(X1,Y), punto(X1,Y1)), Segmentos),
    member(segmento(punto(X1,Y1), punto(X,Y1)), Segmentos),
    member(segmento(punto(X,Y1), punto(X,Y)), Segmentos),
    ((horizontal(segmento(punto(X,Y), punto(X1,Y))),
    vertical(segmento(punto(X1,Y), punto(X1,Y1))));
    (horizontal(segmento(punto(X1,Y), punto(X,Y1))),
    vertical(segmento(punto(X,Y1), punto(X,Y))))), !.
```

Para determinar si es un rectángulo, se verifica que existan 4 segmentos, se verifica el polígono sea cerrado y rectificamos que hayan cuatro puntos unidos entre sí, por ejemplo, el segmento 1 se conforma de los puntos (X,Y) y (X1,Y), para después ir verificando los otros 3 segmentos que se van conectando con el segundo punto del segmento anterior, y a la vez se verifica que 2 segmentos sean horizontales y 2 segmentos sean verticales que de igual forma estén conectados entre sí, un vertical con dos horizontales, dándonos así como resultado un cuadrado.

```
rectangulo([segmento(punto(1,1), punto(1,3)), segmento(punto(1,3), punto(3,3)), segmento(punto(3,3), punto(3,1)),
segmento(punto(3,1), punto(1,1))]).
true
```

El código se puede verificar en el siguiente enlace:

<https://drive.google.com/file/d/1noOo6MC2y1AsWCxF1gww6urnh27aPTej/view?usp=sharing>

Cabe aclarar que por algún motivo que no sabemos si la estructura de segmento está definida como `segmento(punto(X1,Y1),punto(X2,Y2))`. Nos va a soltar la advertencia de Singleton variables, por lo tanto decidimos para que no muestre la advertencia lo escribimos de la siguiente manera `segmento(punto(_,_),punto(_,_))`.