



---

*Dpto. Ingeniería Electrónica, de Sistemas Informáticos y Automática*  
*Área de Ciencias de la Computación e Inteligencia Artificial*  
**Programación Declarativa**  
**Corrección de los exámenes finales**

---

## **Teoría:**

### **Test:**

1: C, 2: C, 3: B, 4: B, 5: A, 6: D

### **Ejercicio 1:**

% Solucion 1

```
s([0,0]).  
s([1,1]).  
s([A,B|Rest]):-  
    sumabinaria(A,B,C),  
    s([C|Rest]).
```

% Solucion 2

```
s([]).  
s([0]).  
s([A,B|Rest]):-  
    sumabinaria(A,B,C),  
    s([C|Rest]).
```

% Solución 3.1

```
suma([],0).  
suma([C1|R1],X):-  
    suma(R1,X1),  
    sumabinaria(C1,X1,X).
```

% Definicion

```
redun(Lista):-  
    reverse(Lista,[Ultimo|Resto]),  
    suma(Resto,Ultimo).
```

% Solucion 3.2

```
redun2(Lista):-  
    suma(Lista,0).
```



**Ejercicio 2:**

a)

paralelo( serie( R1, R2 ) , serie( R3, paralelo( R4,R5 ) ) ).

O bien

paralelo( serie( 1000, 1000 ) , serie( 1000, paralelo( 2000, 1000 ) ) ).

b)

resistencia(serie(R1,R2), Rt) :-  
resistencia(R1, Rt1), resistencia(R2, Rt2),  
Rt is Rt1 + Rt2.

resistencia(paralelo(R1,R2), Rt) :-  
resistencia(R1, Rt1), resistencia(R2, Rt2),  
Rt is 1/ ( 1/Rt1 + 1/Rt2 ).

resistencia(R,R).

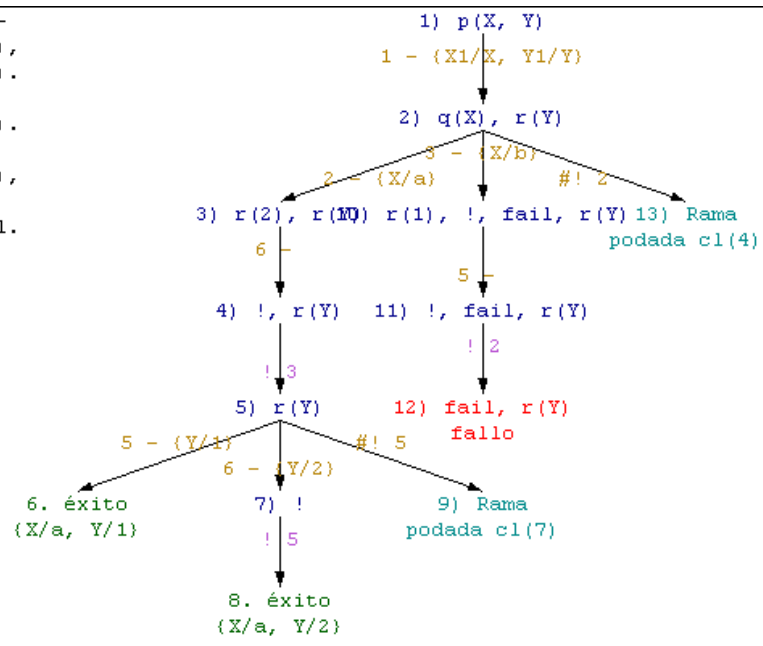
objetivo: resistencia(paralelo(serie(1000,1000),serie(1000,paralelo(2000,1000))), Rt ).

**Ejercicio 3:**

```

1 p(X, Y):-
  q(X),
  r(Y).
2 q(a):-
  r(2).
3 q(b):-
  r(1),
  !,
  fail.
4 q(_A):-
  !.
5 r(1).
6 r(2):-
  !.
7 r(3).

```





---

*Dpto. Ingeniería Electrónica, de Sistemas Informáticos y  
Automática*  
*Área de Ciencias de la Computación e Inteligencia Artificial*  
**Programación Declarativa**  
**Corrección de los exámenes finales**

---

**Ejercicio 4:**

**% apartado a**

bloque(a).  
bloque(b).  
bloque(c).

inicial( [  
    sobre(suelo,bloque(a)),  
    sobre(suelo,bloque(b)),  
    sobre(suelo,bloque(c))  
] ).

final( [  
    sobre(bloque(c),bloque(a)),  
    sobre(suelo,bloque(c)),  
    sobre(bloque(b),bloque(c))  
] ).

**% apartado b**

libre( Bloque, Estado ):-  
    not( esta(sobre(Bloque,\_),Estado) ).

apilar(A,B,E1,E2):-  
    A \= B,  
    libre(A,E1),  
    libre(B,E1),  
    eliminar(sobre(\_,A),E1,Etmp),  
    insertar(sobre(B,A),Etmp,E2).

desapilar(A,E1,E2):-  
    libre(A,E1),  
    not( esta(sobre(suelo,A),E1) ),  
    eliminar(sobre(\_,A),E1,Etmp),  
    insertar(sobre(suelo,A),Etmp,E2).



---

*Dpto. Ingeniería Electrónica, de Sistemas Informáticos y  
Automática*  
*Área de Ciencias de la Computación e Inteligencia Artificial*  
**Programación Declarativa**  
**Corrección de los exámenes finales**

---

**Ejercicio 5:**

a)

multiplos :: Int -> [Int] -> [Int]  
multiplos a x = [y | y <- x, mod y a == 0]

b)

potencia :: Float -> Int -> Float  
potencia x 1 = x  
potencia x y  
| y >= 1 = x \* potencia x (y-1)

o bien

potencia :: Float -> Int -> Float  
potencia x 1 = x  
potencia x y = x \* potencia x (y-1)