

### Algoritmo FWC normal (sin Metas)

```
void Fwc_Pasada(R[1..N], BH) { //Realiza una pasada a la Base de Conocimientos, R[1..N]
    for i=1 to N do{
        if (!marcada(R[i]) && R[i]  $\Rightarrow$  c){
            BH.add(c);
            marcar(R[i]); //Marcar la regla R[i], para evitar que ésta vuelva a disparar en la siguiente pasada.
        }
    }
}
```

```
void FWC(R[1..N], BH) { // FWC sin metas.
    DesmarcarReglas();

    while (true){
        int L=BH.lenght();

        Fwc_Pasada(R, BH);

        if (L == BH.length()) //La BH no creció. Salir.
            break;
    } //End while

    MostrarLosNuevosHechosEncontrados();
}
```

## Algoritmo FWC para una Meta

```
void Fwc_Pasada(R[1..N], BH, Meta) { //Hacer una pasada a la Base de Conocimientos, R[1..N]
    for i=1 to N do{
        if (!marcada(R[i]) && R[i]  $\Rightarrow$  c){
            BH.add(c);
            marcar(R[i]); //Marcar la regla R[i], para evitar que ésta vuelva a disparar en la siguiente pasada.

            if (BH.contains(Meta)) //La Meta se unificó. Salir de la pasada.
                break;
        }
    }
}
```

```
void FWC(R[1..N], BH, Meta) { // FWC para una Meta
    DesmarcarReglas();

    while (true){
        int L = BH.lenght();

        Fwc_Pasada(R, BH, Meta);

        if (L == BH.length() || BH.contains(Meta))
            break; //La BH no creció o la Meta se unificó. Salir.

    } //End while

    if (BH.contains(Meta))
        println(Meta + "=" + valor);
    else
        println("No se encontraron conclusiones para " + Meta);
}
```

## Algoritmo FWC para k Metas

```
void Fwc_Pasada(R[1..N], BH, Lista Meta) { //Hacer una pasada a la BC, R[1..N]
    for i=1 to N do{
        if (!marcada(R[i]) && R[i] | $\Rightarrow$  c){
            BH.add(c);
            marcar(R[i]); //Marcar la regla R[i], para evitar que ésta vuelva a disparar en la siguiente pasada.

            Meta.del(c); //Lo disparado por R[i], puede ser una de las metas. Si es así, quitarla de la Lista Meta.

            if (Meta.Vacia()) //TODAS las metas se unificaron. Salir de la pasada.
                break;
        }
    }
}
```

```
void FWC(R[1..N], BH, Lista Meta) { // FWC para K Metas, las cuales están en la Lista Meta
    DesmarcarReglas();

    while (true){
        int L = BH.lenght();

        Fwc_Pasada(R, BH, Meta);

        if (L == BH.length() || Meta.Vacia())
            break; //La BH no creció o TODAS las metas se unificaron. Salir.

    } //End while

    MostrarLasMetasUnificadas();
}
```