

Lógica

Finales

07/07/03 – Ejercicio 1

```
listaEmpresas([[E,F3]|Xs],Aux,M1,M2) :- relacion(M1,E,F1,F2),relacion(M2,E,F3,F4),
    diferenciaFecha(F2,F3,Dias),Dias =< 15,
    not(miembro([E,F3],Aux)),!,concatenar(Aux,[[E,F3]],Aux2),listaEmpresas(Xs,Aux2,M1,M2).
listaEmpresas([],Aux,M1,M2).
```

```
estudioEmpleo(NombreMujer1,NombreMujer2,ListaEmpresas) :-
listaEmpresas(ListaEmpresas,[],NombreMujer1,NombreMujer2).
```

09/02/04 – Ejercicio 1

```
subSecuenciaRestringida(L, R, S) :- subSec(L,S), sumaElem(S,T), R is T.
```

28/02/05 – Ejercicio 2

```
suma_antteriores(L,N) :-
nPrimerosT(L,L1),long(L1,N),sacarUltimo(L1,L2),sumarElem(L2,S),enesimoElem(L,N,S).
```

Sin Fecha 2 – Ejercicio 2

```
listaResultados([R|Rs],Aux,Equipo) :-
partido(partido(Num),local(Equipo),V,golesLocal(Gl),golesVisitante(Gv)),
    not(miembro(Num,Aux)),!,concatenar(Aux,[Num],Aux2),R is Gl -
Gv,listaResultados(Rs,Aux2,Equipo).
```

```
listaResultados([R|Rs],Aux,Equipo) :-
partido(partido(Num),L,visitante(Equipo),golesLocal(Gl),golesVisitante(Gv)),
    not(miembro(Num,Aux)),!,concatenar(Aux,[Num],Aux2),R is Gv -
Gl,listaResultados(Rs,Aux2,Equipo).
```

```
listaResultados([],_Aux,_Equipo).
```

```
listaJugados([Num|Js],Aux,DNI) :- juega(id(DNI),partido(Num)),
    not(miembro(Num,Aux)),!,concatenar(Aux,[Num],Aux2),listaJugados(Js,Aux2,DNI).
```

```
listaJugados([],_Aux,_DNI).
```

```
partidosGanados([],0).
```

```
partidosGanados([X|Xs],T) :- partidosGanados(Xs,T1),X > 0,!,T is T1 + 1.
```

partidosGanados([X|Xs],T) :- partidosGanados(Xs,T).

numeroPartidosGanados(Equipo,N) :- listaResultados(R,[],Equipo),partidosGanados(R,N).

partidosJugados(DNI,LP) :- listaJugados(LP,[],DNI).

24/02/03 – Ejercicio 2 (hay que probarlo)

armarPoligonal(_Pi, _Pf, [], []):- !.

armarPoligonal([Xi, Yi], [Xf, Yf], [X, Y|RestoPuntos], S):- armarPoligonalAux([Xi, Yi], [Xf, Yf], [X, Y|RestoPuntos], [], S).

armarPoligonalAux([X, Y], [X, Y], _L, _A, S):- !.

armarPoligonalAux([Xi, Yi], [Xf, Yf], [Px, Py|Ps], Aux, [S|Ss]):-

not(miembro([Xi, Yi], Aux)), !, not(interseccion(Xi, Yi, Px, Py)),

concatenar([Xi, Yi], Aux, Aux2), concatenarElem([Xi, Yi], [], S).

eliminaMx([Px, Py], [Px, Py|Ps], Salida), armarPoligonal([Xi, Yi], [Xf, Yf], Salida, Aux2, Ss).

armarPoligonalAux([Xi, Yi|Pis], [Xf, Yf|Pif], [Px Py|Ps], Aux, [S|Ss]):-

not(miembro([Xi, Yi], Aux)), !, interseccion(Xi, Yi, Px, Py),

armarPoligonal([Xi, Yi], [Xf, Yf], Ps, Aux, [S|Ss]).

Parcial 2004 – Ejercicio

par(0).

par(X) :- Z is X-1, Z > 0, impar(Z).

impar(X) :- Z is X-1, par(Z).

sumaPar_Impar(L,Spar,Simpar) :- sumaPar_Impar2(L,0,Spar,0,Simpar).

sumaPar_Impar2([X|Xs],Spar,Spar,SimparIn,SimparOut) :- impar(X),SimparOut is SimparIn + X,
sumaPar_Impar2(Xs,Spar,Spar,SimparIn,SimparOut).

sumaPar_Impar2([X|Xs],SparIn,SparOut,Simpar,Simpar) :- par(X),SparOut is SparIn + X,
sumaPar_Impar2(Xs,SparIn,SparOut,Simpar,Simpar).

sumaP_I(L,P,I):-sumaP_I2(L,0,0,P,I).

sumaP_I2([H|T],Pin,Iin,Pout,Iout):-par(H),Pw is Pin+H, sumaP_I2(T,Pw,Iin,Pout,Iout).

sumaP_I2([H|T],Pin,Iin,Pout,Iout):-impar(H),Iw is Iin+H, sumaP_I2(T,Pin,Iw,Pout,Iout).

sumaP_I2([],P,I,P,I).