



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

## TP II: Programación Lógica

---

Paradigmas de Lenguajes de Programación  
Segundo Cuatrimestre de 2017

Integrante	LU	Correo electrónico
Juan Cruz Basso	627/14	jcbasso95@gmail.com
Emanuel Lamela	021/13	emanuel93_13@hotmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

%%% Ejercicio 1 %%%

```
%contenido(+?Tablero, ?Fila, ?Columna, ?Contenido)
contenido(Tablero, Fila, Columna, Contenido) :-
    nth1(Fila, Tablero, FilaTablero),
    nth1(Columna, FilaTablero, Contenido).
```

%%% Ejercicio 2 %%%

```
%disponible(+Tablero, ?Fila, ?Columna)
disponible(Tablero, Fila, Columna) :-
    posicionLibre(Tablero, Fila, Columna),
    forall(adyacenteEnRango(Tablero, Fila, Columna, FilaAdy, ColumnaAdy),
        posicionLibre(Tablero, FilaAdy, ColumnaAdy)).
```

```
%posicionLibre(+?Tablero, ?Fila, ?Columna)
posicionLibre(Tablero, Fila, Columna) :-
    contenido(Tablero, Fila, Columna, Cont),
    Cont \== o.
```

%%% Ejercicio 3 %%%

```
%puedoColocar(+CantPiezas, ?Direccion, +Tablero, ?Fila, ?Columna)
puedoColocar(1, Direccion, Tablero, Fila, Columna) :-
    esDireccion(Direccion),
    disponible(Tablero, Fila, Columna).
puedoColocar(CantPiezas, Direccion, Tablero, Fila, Columna) :-
    CantPiezas > 1,
    disponible(Tablero, Fila, Columna),
    siguientePosicionEnDireccion(Direccion, Fila, Columna, X, Y),
    PiezasRestantes is CantPiezas - 1,
    puedoColocar(PiezasRestantes, Direccion, Tablero, X, Y).
```

```
%esDireccion(?Direccion)
esDireccion(Direccion) :- member(Direccion, [horizontal, vertical]).
```

```
%siguientePosicionEnDireccion(?Direccion, +Fila, +Columna, -SigFila, -SigColumna)
siguientePosicionEnDireccion(horizontal, Fila, Columna, Fila, SigColumna) :-
    SigColumna is Columna + 1.
siguientePosicionEnDireccion(vertical, Fila, Columna, SigFila, Columna) :-
    SigFila is Fila + 1.
```

%%% Ejercicio 4 %%%

```
%ubicarBarcos(+Barcos, +?Tablero)
```

```
ubicarBarcos(Barcos, Tablero) :-
    maplist(ubicarBarco(Tablero), Barcos).

%ubicarBarco(+?Tablero, +Barco)
ubicarBarco(Tablero, Barco) :-
    puedoColocar(Barco, Direccion, Tablero, Fila, Columna),
    colocarBarco(Barco, Direccion, Tablero, Fila, Columna).

%colocarBarco(+PiezasBarco, ?Direccion, +?Tablero, +Fila, +Columna)
/* Este caso se debe evaluar particularmente para no generar repetidos colocando */
/* barcos de tamaño 1. */
/* Este predicado ASUME que se pueden colocar las piezas de los barcos en las */
/* posiciones pedidas. */
colocarBarco(1, horizontal, Tablero, Fila, Columna) :-
    contenido(Tablero, Fila, Columna, o).
colocarBarco(PiezasBarco, Direccion, Tablero, Fila, Columna) :-
    PiezasBarco > 1,
    contenido(Tablero, Fila, Columna, o),
    siguientePosicionEnDireccion(Direccion, Fila, Columna, SigFila, SigColumna),
    PiezasRestantes is PiezasBarco - 1,
    colocarBarco(PiezasRestantes, Direccion, Tablero, SigFila, SigColumna).

%%% Ejercicio 5 %%%

%completarConAgua(+?Tablero)
completarConAgua(Tablero) :-
    maplist(completarFilaConAgua, Tablero).

%completarFilaConAgua(?FilaTablero)
completarFilaConAgua(FilaTablero) :-
    maplist(completarPosicionConAgua, FilaTablero).

%completarPosicionConAgua(?ElementoTablero)
completarPosicionConAgua(~).
completarPosicionConAgua(ElementoTablero) :-
    ElementoTablero == o.

%%% Ejercicio 6 %%%
%golpear(+Tablero, +NumFila, +NumColumna, -NuevoTab)
golpear([FilaGolpear | Filas], 1, NumColumna, [FilaGolpeada | Filas]) :-
    golpearFila(FilaGolpear, NumColumna, FilaGolpeada).
golpear([Fila | Filas], NumFila, NumColumna, [Fila | TableroGolpeado]) :-
    NumFila > 1,
    ProximaFila is NumFila - 1,
    golpear(Filas, ProximaFila, NumColumna, TableroGolpeado).
```

```
%golpearFila(+FilaGolpear, +NumColumna, -NuevaFila).
golpearFila([_ | Elementos], 1, ['~' | Elementos]).
golpearFila([Elemento | Elementos], NumColumna, [Elemento | FilaGolpeada]) :-
    NumColumna > 1,
    ProximaColumna is NumColumna - 1,
    golpearFila(Elementos, ProximaColumna, FilaGolpeada).

%%% Ejercicios 7 y 8 %%%
%atacar(+Tablero, +Fila, +Columna, -Resultado, -NuevoTab)
% Completar instanciación soportada y justificar.
% Tablero: Es reversible si NuevoTab, Fila y Columna están instanciados.
% Fila: Es reversible si Tablero y NuevoTab están instanciados.
% Pincha en golpear por el > 1.
% Columna: Es reversible si Tablero y NuevoTab están instanciados.
% Pincha en golpear por el > 1.
% Resultado: Es reversible si al menos Tablero, Fila y Columna,o
% Tablero y NuevoTab lo están.
% NuevoTab: Es reversible si Tablero, Fila y Columna están instanciados.
atacar(Tablero, Fila, Columna, 'agua', Tablero) :-
    esAgua(Tablero, Fila, Columna).
atacar(Tablero, Fila, Columna, 'hundido', NuevoTab) :-
    golpear(Tablero, Fila, Columna, NuevoTab),
    esBarco(Tablero, Fila, Columna), esAgua(NuevoTab, Fila, Columna),
    forall(adyacenteEnRango(Tablero, Fila, Columna, FilaAdy, ColumnaAdy),
        esAgua(NuevoTab, FilaAdy, ColumnaAdy)).
atacar(Tablero, Fila, Columna, 'tocado', NuevoTab) :-
    golpear(Tablero, Fila, Columna, NuevoTab),
    esBarco(Tablero, Fila, Columna), esAgua(NuevoTab, Fila, Columna),
    adyacenteEnRango(Tablero, Fila, Columna, FilaAdy, ColumnaAdy),
    esBarco(Tablero, FilaAdy, ColumnaAdy).

%esBarco(+Tablero, ?Fila, ?Columna)
esBarco(Tablero, Fila, Columna) :- contenido(Tablero, Fila, Columna, o).

%esAgua(+Tablero, ?Fila, ?Columna)
esAgua(Tablero, Fila, Columna) :- contenido(Tablero, Fila, Columna, ~).
```