

# Sistemas Inteligentes

## Práctica 1

Memoria: Problema de las N Reinas

Esteban Puentes Silveira

## Código entregado

```
nqueens(Queens,Size) :-  
    field(Size,T,Du,Dv),  
    ocupate(Queens,T,T,Du,Dv).
```

```
field(N,T,Du,Dv) :-  
    NegN is -N,  
    DoubleN is 2*N,  
    getdim(1,N,T),  
    getdim(NegN,N,Du),  
    getdim(2,DoubleN,Dv).
```

```
getdim(Max, Max, [Max]).  
getdim(Min, Max, [Min | Result]) :-  
    Min =< Max,  
    N1 is Min+1,  
    getdim(N1, Max, Result).
```

```
ocupate([],[],ListFila,Diag1,Diag2).  
ocupate([Y|Fila],[X|Columna],ListFila,Diag1,Diag2) :-  
    remove(Y,ListFila,ListFila1),  
    V is X-Y,  
    remove(V,Diag1,Diag1_1),  
    W is X+Y,  
    remove(W,Diag2,Diag2_1),  
    ocupate(Fila,Columna, ListFila1,Diag1_1,Diag2_1).
```

```
remove(X,[X|Cdr],Cdr).  
remove(X,[Cat|Cdr],[Cat|Cdr1]) :-  
    remove(X,Cdr,Cdr1).
```

## Análisis

**Método principal para la ejecución del programa, esta crea el tablero y calcula las posiciones de las reinas, Queens es la variable de salida donde se obtienen en una lista las posiciones en Y donde se colocará cada reina en el tablero, y la variable Size que indica el tamaño del tablero**

```
nqueens(Queens,Size) :-  
    field(Size,T,Du,Dv),  
    ocupate(Queens,T,T,Du,Dv).
```

**Este es el método que genera el tablero, N indica el tamaño que tendrá el tablero, T, Du y Dv son variables de salida, T es una lista desde 1 hasta el tamaño del tablero, que represente las filas y las columnas; Du es una lista donde cada elemento representa el valor X-Y de las casillas de una diagonal (Cada elemento representa una diagonal); Dv X+Y de las casillas de una diagonal (Cada elemento representa una diagonal).**

```
field(N,T,Du,Dv) :-  
    NegN is -N,  
    DoubleN is 2*N,  
    getdim(1,N,T),  
    getdim(NegN,N,Du),  
    getdim(2,DoubleN,Dv).
```

**Método que genera una lista que contiene un rango determinado de números. Es utilizado para crear las listas con las dimensiones del tablero, tanto para filas y columnas, como diagonales. Las variables Min y Max representan el valor mínimo y máximo de la lista respectivamente y la tercera variable es la variable de salida con la lista desde Min hasta Max.**

```
getdim(Max, Max, [Max]).  
getdim(Min, Max, [Min | Result]) :-  
    Min =<= Max,  
    N1 is Min+1,  
    getdim(N1, Max, Result).
```

**Método que calcula las posiciones de las reinas, recibe las variables de entrada Columna, ListaFila, Diag1 y Diag2, que son respectivamente la lista con las posiciones libres en columnas, filas y diagonales. Se ocupa la posición X,Y, se elimina de las listas de libres las posiciones que están en la misma fila, columna y diagonales que la reina colocada y se llama recursivamente al método con las nuevas listas. Este método calcula todas las posibilidades, y genera fallo en las cuáles no sean vacías las listas de Filas y Columnas, por lo que la variable de salida Fila solo devolverá los casos en los que hayan tantas reinas como el tamaño original de estas listas (N).**

```
ocupate([],[],ListFila,Diag1,Diag2).  
ocupate([Y|Fila],[X|Columna],ListFila,Diag1,Diag2) :-  
    remove(Y,ListFila,ListFila1),  
    V is X-Y,  
    remove(V,Diag1,Diag1_1),  
    W is X+Y,
```

```
remove(W,Diag2,Diag2_1),  
ocupate(Fila,Columna, ListFila1,Diag1_1,Diag2_1).
```

**Método que elimina el valor X de una lista y devuelve la nueva lista sin dicho elemento.**

```
remove(X,[X|Cdr],Cdr).
```

```
remove(X,[Cat|Cdr],[Cat|Cdr1]) :-
```

```
    remove(X,Cdr,Cdr1).
```

## Apartado Soluciones

Para N igual a 8 el programa descrito encuentra 92 soluciones de un total de 92 soluciones.