



---

# **Especificación tercer proyecto**

**tema**

**Benchmarking P vs NP: NReinas**

**Versión 1.0**

**Valor 10 %**

**Preparado por Eddy Ramírez**

---



# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Propósito . . . . .	5
1.2. Alcance . . . . .	5
1.3. Referencias . . . . .	5
<b>2. Descripción general</b>	<b>6</b>
2.1. Entrada del programa . . . . .	6
2.1.1. Prolog . . . . .	6
2.1.2. Erlang . . . . .	7
2.2. GUI . . . . .	7
2.3. Aspectos generales . . . . .	7
2.3.1. Evaluación . . . . .	7
2.3.2. Aspectos técnicos . . . . .	8
2.3.3. Aspectos administrativos . . . . .	8

# 1 Introducción

Este documento contiene la especificación del tercer proyecto programado para el curso de Lenguajes de Programación.

## 1.1. Propósito

El proyecto tiene como propósito dar al estudiante familiaridad con un lenguaje funcional como Erlang, utilizar el backtracking de prolog y también permitir que el estudiante profundice en la comparación de soluciones P y NP para el caso particular de las NReinas.

## 1.2. Alcance

Se espera que el estudiante desarrolle un sistema capaz de encontrar una solución para el problema de las NReinas.

## 1.3. Referencias

1. <http://www.uv.es/asepuma/X/J24C.pdf>
2. <http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf>
3. <http://sabia.tic.udc.es/mgestal/cv/aaggtutorial/tutorialalgoritmosgeneticos.pdf>
4. <http://www.youtube.com/watch?v=YhjPmu9Qddg>
5. [http://es.wikipedia.org/wiki/Problema\\_de\\_las\\_ocho\\_reinas](http://es.wikipedia.org/wiki/Problema_de_las_ocho_reinas)
6. Mitchel, Melanie. *An Introduction to Genetic Algorithms*, MIT Press, 1999, ISBN 0-262-13316-4 (HB)

## 2 Descripción general

Se debe de hacer una comparación tanto en facilidad de escribir código como en claridad de la solución, pero también en el  $O(n)$  que implica encontrar 1 solución, varias soluciones o todas las soluciones entre dos aproximaciones clásicas.

La primera aproximación es la implementación por backtracking, para ello se utilizará el lenguaje de programación Prolog, el cual debe de ser capaz de encontrar todas las soluciones posibles para un problema de tamaño  $n$  cada vez que se oprima la opción *or* (;).

La segunda solución consiste en la solución por medio de algoritmos genéticos. En esta aproximación se utilizará el lenguaje de programación Erlang. Esta tendrá tres modalidades, la primera consiste en dada una población inicial aleatoria (pero válida para ser solución), generará a lo sumo  $k$  generaciones, mostrando el mejor de cada generación, si encuentra una solución, entonces se detiene. La segunda forma es que no se detenga hasta encontrar una solución. La tercera forma es que se puedan hacer hilos para llamar a cualquiera de las dos versiones anteriores. Nótese que si se llaman a  $h$  hilos con la segunda opción, en principio puedo obtener hasta  $h$  soluciones diferentes.

### 2.1. Entrada del programa

Dado que en realidad habrán dos programas, uno en Prolog y otro en Erlang, a continuación se describen las especificaciones de cada uno de los programas

#### 2.1.1. Prolog

En Prolog, todo son relaciones, por lo que la relación debe llamarse  $nq^1$ , la cual recibe dos parámetros, el primero es el  $n$  o tamaño del tablero, la segunda es una matriz de caracteres para representar el tablero, se utilizará el `_` (guión bajo) para señalar un escaque o casilla vacío y la letra `q` (en minúscula o mayúscula) para representar una reina. Los caracteres además se separan por un espacio o una coma (a gusto del programador).

**Ejemplo:**

```
nq(4,X) .
```

```
X = ["_Q__", "___Q", "Q___", "__Q_"] ;  
X = ["__Q_", "Q___", "___Q", "_Q__"]  
yes
```

---

<sup>1</sup>Viene de N-Queen, o N-Reinas en inglés

### 2.1.2. Erlang

En Erlang, al ser funcional, la función  $nq$ <sup>2</sup> recibe 4 parámetros:

1.  $n$   
El tamaño del tablero (entero)
2.  $pob$   
El tamaño de la población (entero)
3.  $mut$   
La probabilidad de mutación (flotante)
4.  $param$  : (Tupla) En este caso, el tamaño de la tupla indicará el tipo de funcionamiento que debe de tener.  
  
    {limit,Generaciones,si}: En este caso, va a funcionar únicamente con generaciones limitadas, si encuentra una solución, entonces se detiene, pero en caso contrario se debe detener al cumplir el número de generaciones indicado. El tercer parámetro es si se debe aplicar o no, elitismo.  
  
    {unlimit, no}: En este caso, va a funcionar hasta encontrar una solución, el segundo parámetro es para saber si va a haber elitismo o no.  
  
    {hilos, Cant,TuplaParams} En este caso funciona con hilos, se van a hacer Cant hilos con los mismos parámetros que tienen el tercer elemento de la tupla, que es a su vez una tupla con los parámetros anteriores.

Cada función deberá desplegar una matriz, similar a la de  $prolog$ <sup>3</sup>.

## 2.2. GUI

## 2.3. Aspectos generales

### 2.3.1. Evaluación

A continuación se muestra una tabla de evaluación sobre los diferentes puntos a tomar en cuenta

---

<sup>2</sup>ídem

<sup>3</sup>Pueden haber cambios de línea entre filas de la matriz

Rubro	Valor				
Definición de los individuos	5 %	-1 % si valorarlos tiene costo superior a $O(n) = n^2$	-5 % si no representa todo el problema		
Definición de la función objetivo	10 %	-2 % por un $O(n)$ superior a $n^2$	5 % si no concuerda el objetivo con la función		
Definición de los otros atributos (cruce, mutación, etc)	5 %	-2 % si no aprovechan las propiedades de AG	-5 % si no cumple con 3 o más propiedades		
Funcionamiento del programa	70 %	35 % por programa en prolog	35 % por programa en Erlang		
Documentación	10 %				

La documentación consiste en tres cuadros comparativos que comparen ventajas y desventajas:

1. P vs NP
2. Backtracking vs Genéticos
3. Prolog vs Erlang para nreinas.

### 2.3.2. Aspectos técnicos

1. Toda la programación debe de realizarse sobre GNU/Linux
2. El uso de variables está absolutamente prohibido. Esto implica que todo define debe definir un  $\lambda$ ,
3. En caso de desear incorporar GUI las bibliotecas que se utilicen deben ser entregadas junto a la tarea.

### 2.3.3. Aspectos administrativos

1. Se debe entregar sólo el código fuente y las bibliotecas en caso de usar GUI.
2. Cualquier sospecha de fraude implica una nota de cero y se aplicarán las sanciones respectivas
3. El no acatar alguna de las indicaciones de los Aspectos Técnicos, se penalizará con nota de cero.
4. El proyecto se entrega junto con los demás proyectos en la última semana de mayo (a más tardar el cuatro de junio)