

# Proyecto funcional

Lenguajes de Programación  
Instituto Tecnológico de Costa Rica  
Eddy Ramírez  
II-Semestre 2020

## 1. El problema de las $n$ Reinas

El popular problema de las  $N$ Reinas es colocar  $n$  reinas en un tablero de ajedrez de  $n \times n$  de tal manera que ninguna de las reinas quede atacando a otras.

Para resolverlo tradicionalmente, se debe hacer mediante *backtracking* o por medio de algún algoritmo heurístico, que aunque normalmente tiene un costo menor en tiempo, no sirve para encontrar todas las soluciones.

En esta imagen se muestra una posible solución de 5-Reinas.

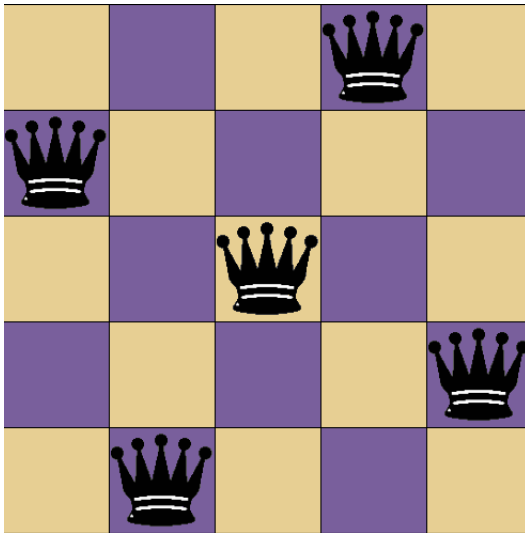


Figura 1: Solución de 5 Reinas

## 2. Especificación del proyecto

Su proyecto consiste en implementar en Scheme y Erlang dos soluciones, una en cada uno. La primera basada en algoritmos genéticos y la segunda basada en Backtracking. Cual se utilice en cada lenguaje, queda a criterio de cada grupo de trabajo.

## 3. Funcionamiento del programa utilizando algoritmos genéticos

Utilizando algoritmos genéticos, se debe de encontrar una solución para un problema de  $n$ -reinas. El programa deberá mostrar al individuo elite de cada generación en pantalla (aunque cambie muy frecuentemente) y debe de trabajar con elitismo (el mejor de una población, debe de pasar a la siguiente).

El porcentaje de mutación lo puede definir cada equipo, pero debe ser menor o igual a 5%. El tamaño de la población es también a criterio de cada grupo (aunque puede quedar en función del número de reinas).

El programa debe detenerse cuando encuentre un individuo en el que ninguna de las reinas ataque a otra.

La función debe llamarse *geneticosNReinas* y su dominio son naturales mayores que 3.

Ejemplo:

```
(geneticosNReinas 5)  
o  
proyecto:geneticosNReinas(4).
```

## 4. Funcionamiento del programa con backtracking

Utilizando una pila (para backtracking), su programa debe hallar soluciones al problema de NReinas y se debe especificar un número particular de soluciones.

La función principal debe llamarse *backtrackingNReinas* y su dominio son dos números naturales,  $a$  y  $b$  tal que  $a > 3$  y  $b > 0$ . Siendo  $a$  el número de reinas por colocar en un tablero de  $a \times a$  y  $b$  el número de soluciones que se espera encontrar. Si se pide un  $b$  mayor que el número total de soluciones, se debe mostrar sólo el número total de soluciones.

Ejemplo:

```
(backtrackingNReinas 5 3)
o
proyecto:backtrackingNReinas(4,2).
```

## 5. Documentación

Todo proyecto debe venir con la documentación en L<sup>A</sup>T<sub>E</sub>X que conste de:

- Portada

Con el logo del TEC, título del proyecto, nombre del curso, nombre de los estudiantes, nombre del profesor y fecha.

- Resumen ejecutivo

En una página (no más, no menos) un resumen del proyecto, su solución y los resultados de las pruebas

- Introducción

Una descripción del documento y en qué consistía el proyecto.

- Marco teórico

Una descripción de los lenguajes utilizados (antecedentes, historia, usos), una breve documentación de las bibliotecas utilizadas (si las hubo) y de los ide's utilizados.

- Descripción de la solución

Detalle del algoritmo de solución para cada caso (backtracking y genéticos) y problemas encontrados durante el desarrollo del proyecto y su respectiva solución.

- Resultados de pruebas

Una vez finalizado el proyecto, debe de tener realizados diversos casos de prueba, tomar tiempo de ejecución (usando el time de cada lenguaje) y mostrar gráficas del contraste de resultados utilizando cada posible solución. (No se esperan menos de 10 contrastes en casos de prueba para cada algoritmo)

- Conclusiones

Un análisis de las gráficas y del proyecto en general. Así como una explicación teórica de los resultados.

- Aprendizajes

De manera individual, qué aprendió cada integrante de los equipos con la realización de este proyecto, a la luz de los objetivos del curso.

- Bibliografía (en formato IEEE)

Se rebajará un punto por cada falta ortográfica presente en la documentación. Si el documento no tiene bibliografía o no está hecho en Latex, se otorgará una nota de cero.

## 6. Rúbrica

Se asignarán dos rúbricas, una para la parte programada  $p$  y otra para la documentación  $d$ . La nota del proyecto será el promedio geométrico de ambas notas ( $\sqrt{p \times d}$ ).

### 6.1. Programas funcionales

Producto	Valor
Respeto al paradigma de programación	20 %
Algoritmo genético	40 %
Backtracking	40 %
<b>Total</b>	100 %

Tener presente que el respeto al paradigma de programación, debe mantenerse a lo largo de todo el proyecto, en Scheme se mantiene la prohibición de las siguientes funciones:

- while
- set!
- if

Mientras que en Erlang se prohíbe el uso de *if*, *while* y *for*.

## 6.2. Documentación

Tener presente que la ortografía resta 1 punto por cada falta y no tener bibliografía implica una nota de cero, lo mismo que no hacerla en L<sup>A</sup>T<sub>E</sub>X.

Producto	Valor
Portada	5 %
Resumen ejecutivo	10 %
Introducción	5 %
Marco teórico	10 %
Descripción de la solución	20 %
Resultados de pruebas	35 %
Conclusiones	10 %
Aprendizajes	5 %
<b>Total</b>	100 %

## 7. Aspectos Técnicos

- El trabajo debe de ser realizado de forma individual, en parejas o tríos.
- Toda la programación debe realizarse en Scheme o Erlang, según corresponda y será revisado sobre Linux.
- **Todas las funciones deben tener un comentario antes del nombre de la misma indicando lo que recibe como parámetro y lo que retorna. (Dominio y Codominio)**
- Los nombres de los archivos deben ser las tres primeras letras de los apellidos de los integrantes. Ejemplo: *ram.scm* o *ram.erl*

## 8. Aspectos Administrativos

- La fecha de entrega de este proyecto es el 18 de octubre de 2020 a las 11:59 pm
- El medio de entrega será por correo electrónico a [edramirez@itcr.ac.cr](mailto:edramirez@itcr.ac.cr)
- El asunto del correo debe ser: 2020-02 IC-4700 Proyecto Funcional [Nombre del estudiante(s)]<sup>1</sup> y debe enviarse con copia a todos los miembros del grupo.

---

<sup>1</sup>Siendo nombre del estudiante el nombre de cada estudiante, no debe llevar los paréntesis cuadrados