

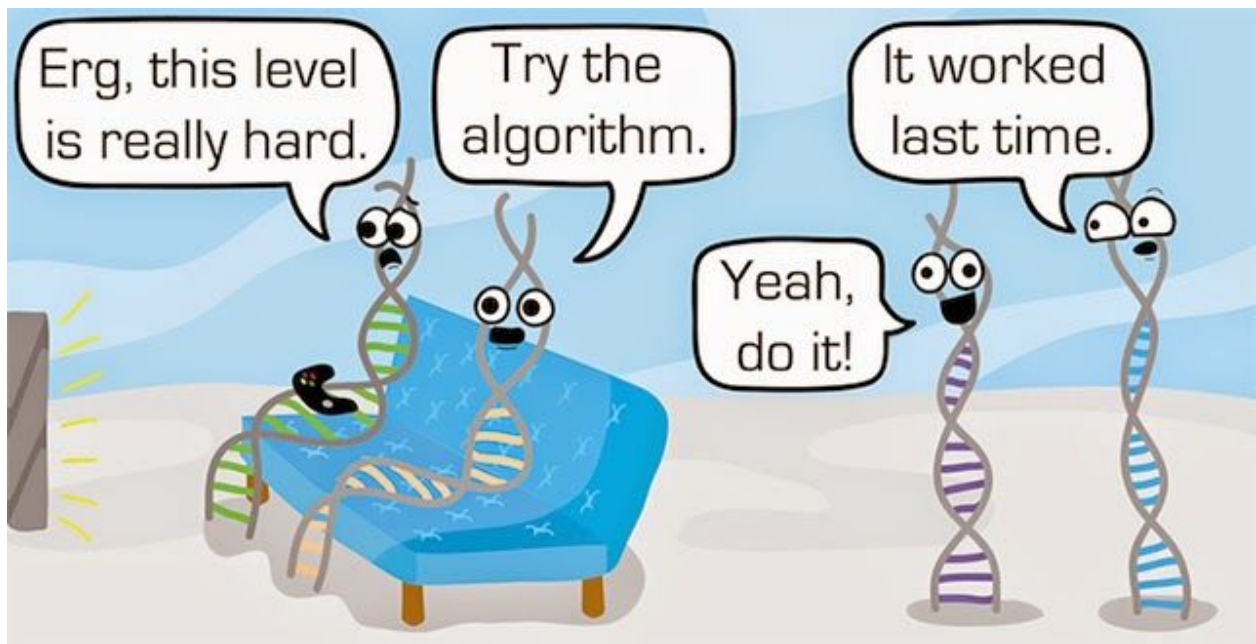
---

## SEMINARIO DE SOLUCION DE PROBLEMAS DE INTELIGENCIA ARTIFICIAL I

**Diego Antonio Vallejo Jiménez**

### Practica 1

Código 211573851



### DESCRIPCIÓN

Una manera muy peculiar e interesante es el uso de algoritmos genéticos para encontrar posibles soluciones óptimas al problema

El algoritmo genético es una técnica de búsqueda que puede usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos.

---

## MARCO TEORICO

### Optimización

Optimizar quiere decir buscar mejores resultados, más eficacia o mayor eficiencia en el desempeño de alguna tarea. De allí que términos sinónimos sean mejorar, optimar o perfeccionar. Mientras que antónimos serían desmejorar o empeorar.

Se dice que se ha optimizado algo (una actividad, un método, un proceso, un sistema, etc.) cuando se han efectuado modificaciones en la fórmula usual de proceder y se han obtenido resultados que están por encima de lo regular o lo esperado. En este sentido, optimizar es realizar una mejor gestión de nuestros recursos en función del objetivo que perseguimos.



Cuando optimizamos un proceso buscamos reducir los costes o maximizar los la eficiencia del proceso o los resultados encontrando los valores de entrada 'Óptimos'.

### Optimización mediante el uso de Algoritmos Genéticos

Una manera muy peculiar e interesante es el uso de algoritmos genéticos para encontrar posibles soluciones óptimas al problema

El algoritmo genético es una técnica de búsqueda que puede usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin.

Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de los mismos.

Hans-Joachim Bremermann (1926-1996) fue el primero en ver la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones con cadenas binarias que se procesaban por medio de reproducción, selección y mutación

---

## DESARROLLO DE LA ACTIVIDAD

El pseudocódigo del algoritmo es:

*Padres* = {Poblacion Generada aleatoriamente}

Mientras( !Criterio de terminacion ) hacer:

    Calcular la condicion para cada padre de la poblacion

*Hijos* <- 0

    Mientras( |*Hijos*| < |*Padres*| ) hacer:

        Usar la seleccion probabilistica de la condición  
        (Ruleta) para seleccionar un par de padres

        Reproducir a los padres para crear a los hijos(*h1,h2*)

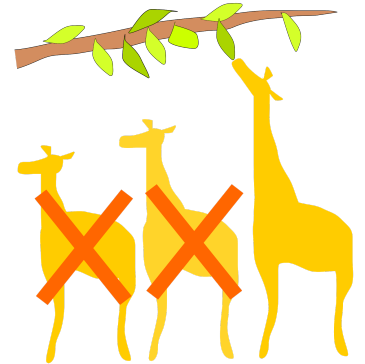
*Hijos* <- *Hijos* + {*h1,h2*}

    Fin Mientras

    Aleatoriamente Mutar algunos Hijos

*Padres* <- *Hijos*

Fin Mientras



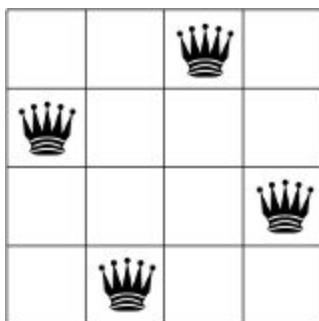
---

## La Problemática a resolver

Resolver el problema de las n reinas con el algoritmo genético

# El problema de N-Reinas

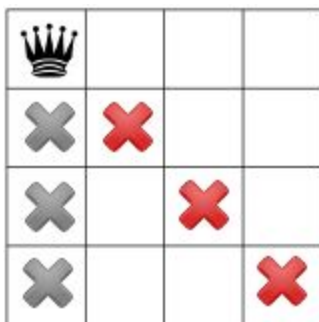
En las siguientes secciones, ilustraremos el solucionador de CP mediante un problema combinatorio basado en el juego de ajedrez. En el ajedrez, una dama puede atacar horizontal, vertical y diagonalmente. El problema de N-queens pregunta:



*¿Cómo se pueden colocar N Reinas en un tablero de ajedrez NxN para que no se ataquen ninguno de los dos?*

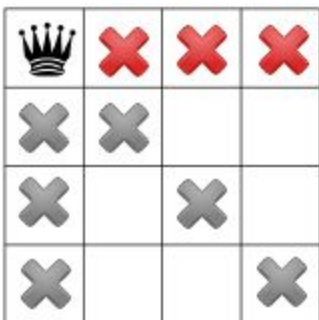
A continuación, puede ver una posible solución al problema N-queens para N = 4.

No hay dos reinas en la misma fila, columna o diagonal.



Una restricción es que solo puede haber una reina en una columna (las X grises debajo), y otra restricción prohíbe dos reinas en la misma diagonal (las X rojas debajo).

Nuestra tercera restricción prohíbe reinas en la misma fila:



Nuestras limitaciones se propagan, podemos probar otra hipótesis y colocar una segunda reina en uno de los cuadrados restantes disponibles. Nuestro solucionador puede decidir colocar en él el primer cuadrado disponible en la segunda columna:

♔	×	×	×
×	×	×	
×	♔	×	
×	×	×	×

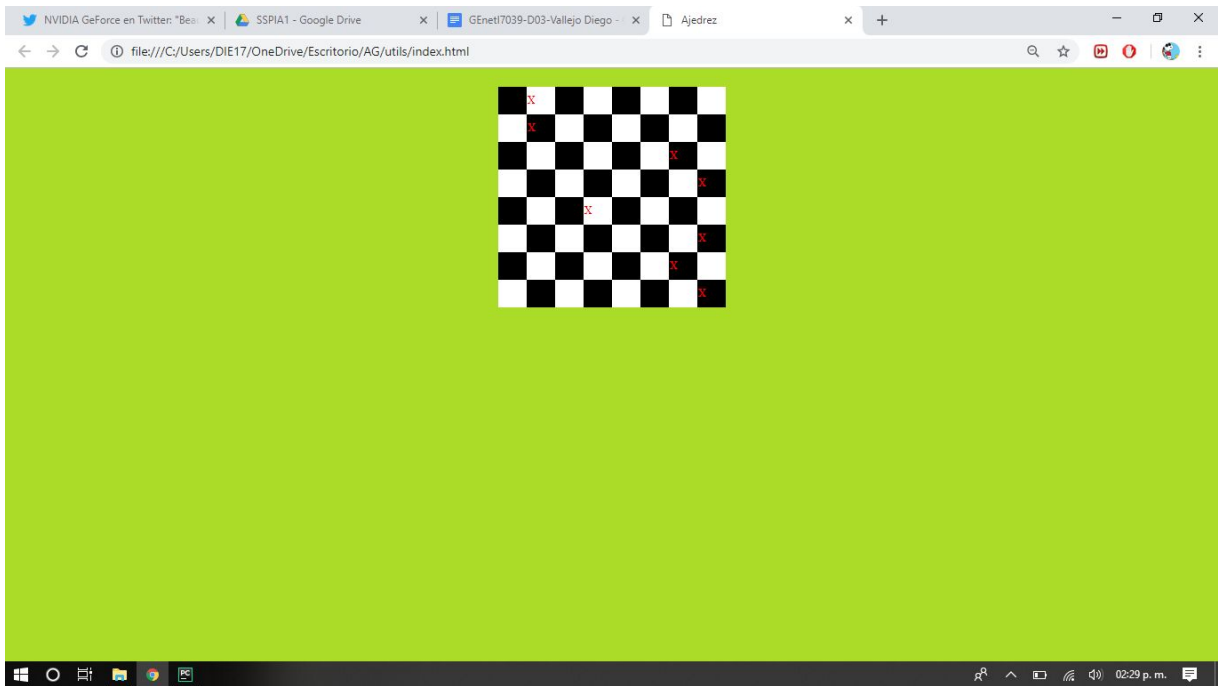
Después de propagar la restricción diagonal, podemos ver que no deja cuadrados disponibles ni en la tercera columna ni en la última fila:

♔	×	×	×
×	×	♔	×
×	×	×	♔
×	♔	×	×

Sin soluciones posibles en esta etapa, debemos retroceder. Una opción es que el solucionador elija el otro cuadrado disponible en la segunda columna. Sin embargo, la propagación de restricciones fuerza a una reina a la segunda fila de la tercera columna, sin dejar puntos válidos para la cuarta dama:

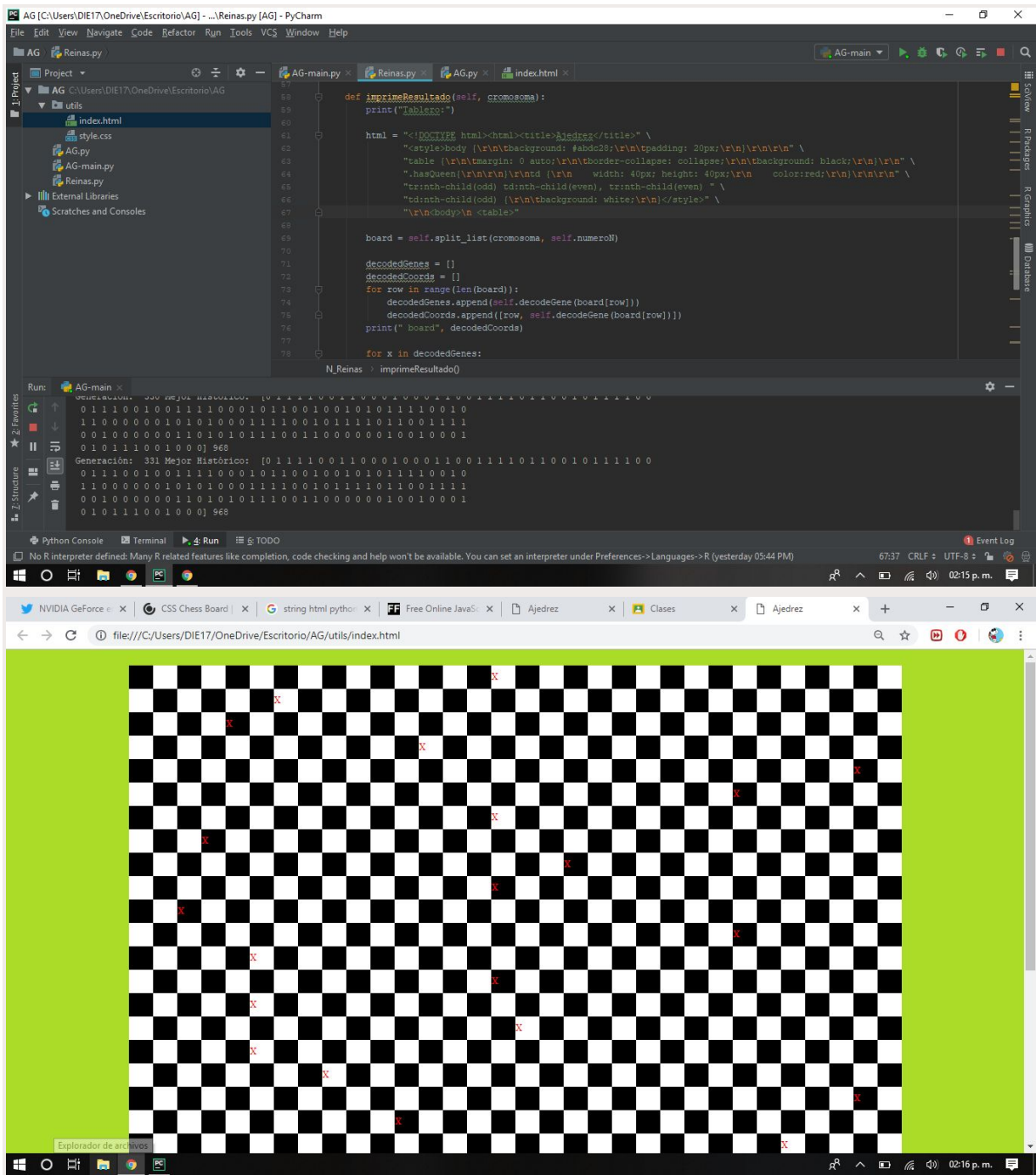
♔	×	×	×
×	×		
×	♔	×	
×	×	×	×

## EJECUCIÓN



Una de las iteraciones dio una curiosa solución





---

## REFLEXION

Esta ha sido una de mis actividades favoritas que he realizado en clases

Personalmente el imitar y representar procesos naturales con software y utilizar los algoritmos para representar o procesar información es un tema que me fascina

Al visualizar este algoritmo me sorprendió su efectividad aunque no siempre encuentra la solución óptima pero se aproximan mucho y demostró ser un método de aproximación interesante y útil para problemas de mayor complejidad.

## REFERENCIAS

<https://developers.google.com/optimization/cp/queens>