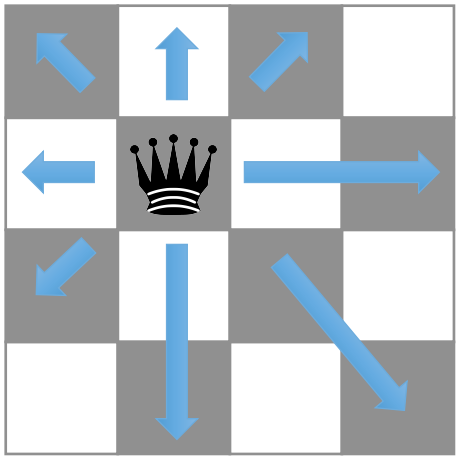
Práctica — Problema de las N-Reinas

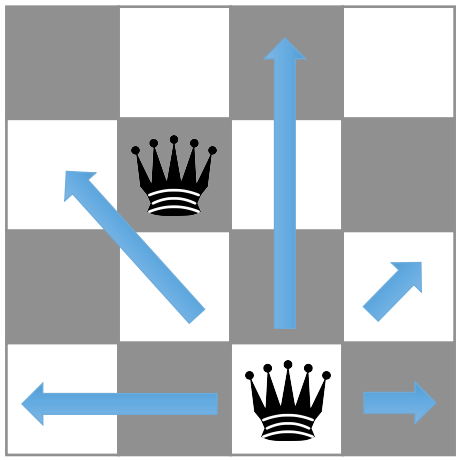
Definición del problema (extraído de Wikipedia)

El problema original de las 8 reinas fue propuesto en 1848 por el ajedrecista alemán Max Bezzel, el cual lo planteó como un pasatiempo. Después del planteamiento original el problema fue tratado por matemáticos como Gauss y Cantor que lo generalizaron a las n-reinas (en un tablero de tamaño nxn). Dijkstra detalló un algoritmo de backtraking para resolver este problema en 1972.

El objetivo del problema es colocar todas las reinas en el tablero sin que se amenacen entre ellas. Recordemos que una reina amenaza a todas las piezas que estén situadas en la misma columna, fila y diagonal que ella. Así en un tablero de 4x4 una reina amenazaría a todas las casillas marcadas en la siguiente imagen:



Con la reina anterior fijada, si quisiéramos situar otra reina en el tablero, sólo podríamos hacerlo en uno de los huecos que quedan, por ejemplo:



Como podemos observar después de haber situado la primera reina en ésa posición podremos colocar sólo dos reinas más, por lo que estas posiciones no se corresponden con una solución del problema.

Objetivos de la práctica

El alumno deberá:

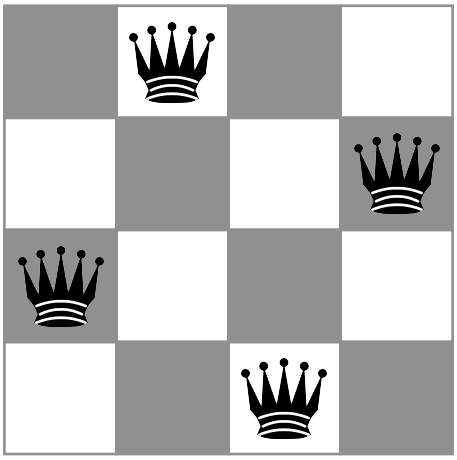
a) Implementar un algoritmo, mediante la técnica de vuelta atrás, que obtenga una solución válida para un tamaño determinado del tablero.
b) Implementar un algoritmo, mediante la técnica de vuelta atrás, que obtenga todas las soluciones posibles para un tamaño determinado del tablero.
c) Evaluar experimentalmente la corrección de su implementación, tal y como se explica posteriormente.

Modelado del problema

Una solución se representará como un array de enteros de tamaño n (para n reinas) de forma que cada entero del array representa la posición (empezando por 0) en la que se sitúa la reina. Por ejemplo una solución para n=4 es:

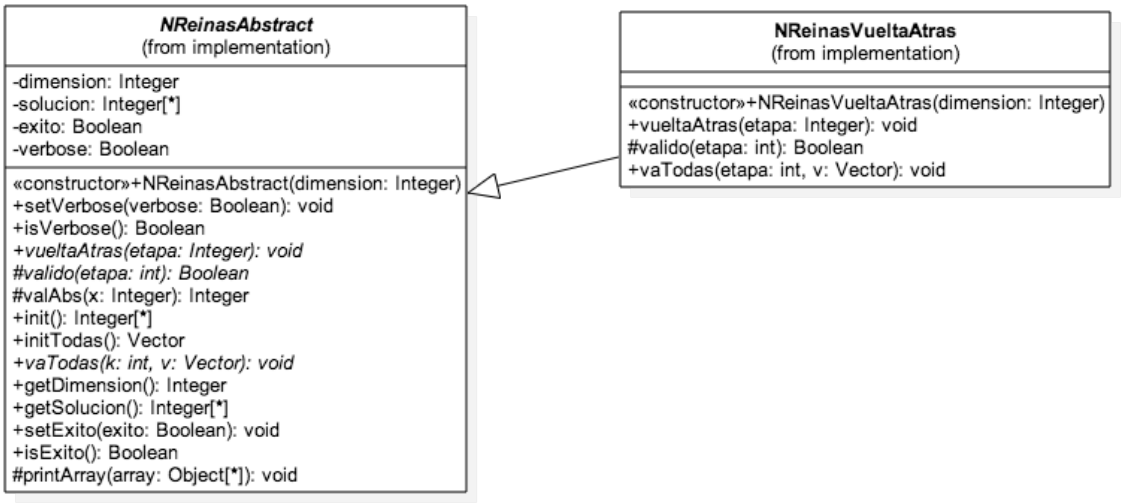
solución = [1, 3, 0, 2]

Cuya representación en el tablero sería la siguiente:



Para la realización de la práctica se proporcionan un conjunto de clases que servirán de apoyo para las pruebas de las implementaciones solicitadas, y para evaluar experimentalmente cómo se comporta el algoritmo para los distintos valores de n.

De todas las clases del proyecto el alumno sólo deberá implementar los métodos marcados con TODO (To-do) de la clase NReinasVueltaAtras que implementa la clase abstracta NReinasAbstract, la cual nos proporciona algunos métodos que pueden ayudarnos en la implementación. La estructura de las clases puede observarse en el diagrama:



Las descripciones de los métodos proporcionados por la clase abstracta son:

- *setVerbose*: selecciona si deseamos que la clase de implementación muestre los mensajes por la consola. Para mostrar mensajes, se proporciona el método *printArray*. De esta manera, el alumno debería imprimir los mensajes que considere oportunos de la siguiente manera:

```
if (isVerbose()) {
        //Mostrar mensaje
}
```

- *isVerbose*: devuelve la selección de mensajes de salida que se ha hecho para la clase.
- *vueltaAtras*: [IMPLEMENTAR] calcula una solución al problema mediante la técnica de vuelta atrás.
- *valido*: [IMPLEMENTAR] devuelve si una solución es válida para una etapa concreta. Simplifica la implementación de los métodos *vueltaAtras* y *vaTodas*.

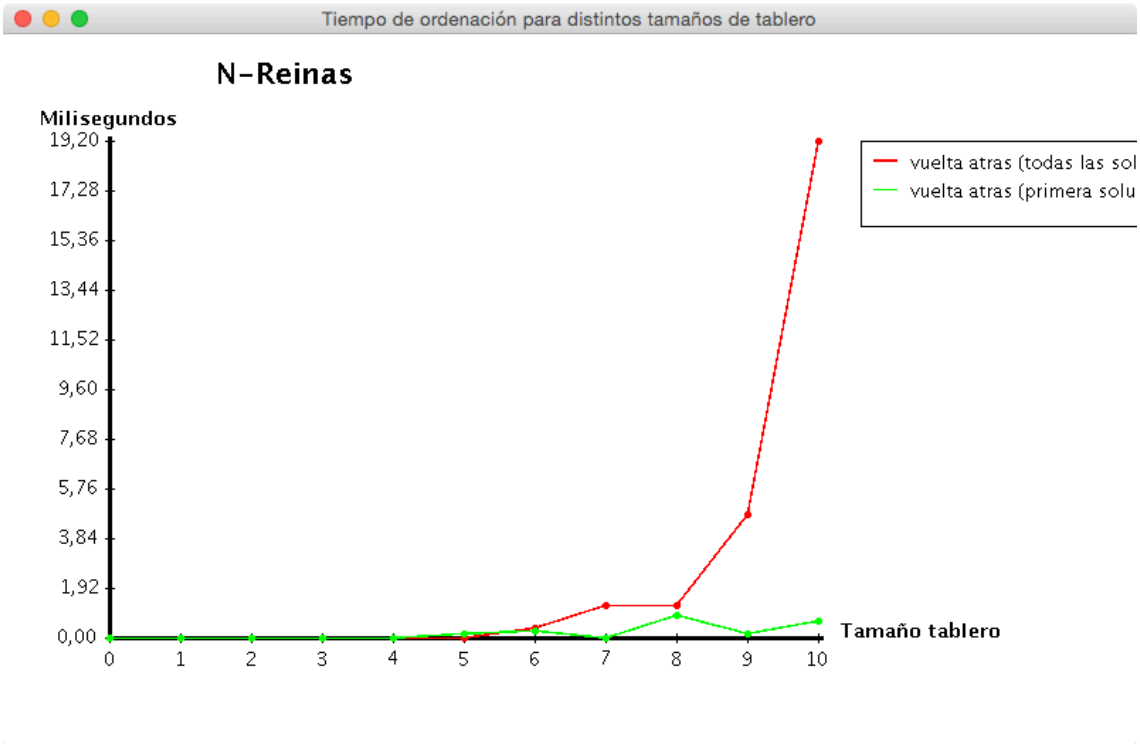
Ejemplo: `nr.valido(5); //Devuelve si la reina en la posición 4 del array es válida con las cuatro reinas anteriores (en las posiciones 0,1,2,3).`

- *vaTodas:* [IMPLEMENTAR] devuelve todas las soluciones válidas mediante la técnica de vuelta atrás.
- *valAbs*: devuelve el valor absoluto de un entero pasado por parámetro.
- *init:* método de entrada para calcular una solución. Se llama desde el driver NReinasDriverFirstSolution. Observar la llamada que realiza para validar que coincide con lo implementado.
- *initTodas:* método de entrada para calcular todas las soluciones. Se llama desde el driver NReinasDriverTodas. Observar la llamada que realiza para validar que coincide con lo implementado.
- *getDimension:* devuelve la dimensión del tablero seleccionada en la clase.
- *getSolucion:* devuelve el valor de la solución actual alojada en la clase.
- *setExito:* informa si se trata de un éxito o no.
- *printArray:* método que imprime por consola el array de objetos pasado por parámetro.

Para abordar la práctica es obligatorio que la clase extienda la clase abstracta NReinasAbstract y que la clase se llame NReinasVueltaAtras. Se recomienda utilizar la plantilla proporcionada.

Validación experimental

Una vez completado el código podemos validar experimentalmente los tiempos de ejecución mediante la clase `GeneradorPruebas`. Esta clase llamará a los drivers correspondientes, los cuales se encargarán de ejecutar la clase implementada, medir los tiempos de ejecución y generar una gráfica con los resultados obtenidos. Por ejemplo esta gráfica se generó con un i5 a 2,5 GHz:



Entrega

Una vez finalizada la práctica debe entregarse el fichero: `NReinasVueltaAtras.java` a través de la actividad SIETTE preparada en el Campus Virtual.

La corrección de este ejercicio se realiza de forma automática validando que el programa es capaz de determinar si una solución parcial es válida, obtener la primera solución y todas las posibles soluciones para distintos tamaños del problema, Para ello se deben superar las pruebas unitarias definidas para JUnit en las clases "`NReinasVueltaAtrasTest*.java`"

Aquellas practicas que no superen ninguna de estas pruebas automáticas serán corregidas manualmente, aunque en ningún caso alcanzaran una calificación superior a 5 sobre 10.