


Recuperatorio	<b>Programación Orientada a Objetos</b> Tecnatura Universitaria en Desarrollo de Videojuegos	
<b>Docente:</b> Ricardo Moran	<b>Alumno:</b> Legajo: Nombre:	<b>Fecha:</b>

### 3. Evaluación Práctica

El objetivo del presente ejercicio es evaluar las capacidades de abstracción y modelado orientado a objetos del estudiante, la aplicación de los conceptos teóricos, y el uso de los recursos prácticos abordados por la asignatura en las unidades incluidas en este examen.

NO será factor en la evaluación la interfaz gráfica, queda a discreción del alumno utilizar la tecnología de presentación de su preferencia dentro de Visual Studio, pero se recomienda a los mismos apegarse a una aplicación en consola para evitar malgastar tiempo y esfuerzo en componentes que no sumarán a la nota de este parcial.

#### 3.1. Restricciones

- Se requiere del alumno realizar una aplicación ejecutable en C# dentro de Visual Studio
- La aplicación desarrollada DEBE compilar.
- Para considerar a una consigna cumplida, la funcionalidad planteada por la misma debe poderse probar, y el código que la provee debe ser orientado a objetos.

#### 3.2. Recomendaciones

Se recomienda al alumno leer con detenimiento las restricciones y las consignas planteadas para comprender exactamente que se pretende de él.

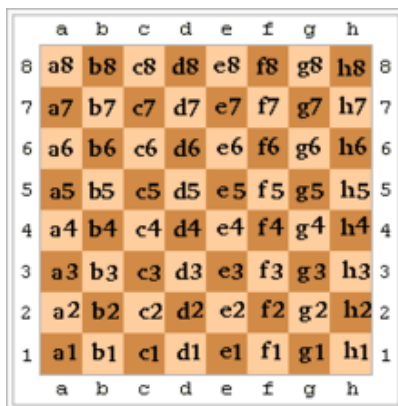
En función de lo solicitado se sugiere analizar detenidamente la descripción del punto 3.3 y modelar las clases y objetos necesarios para cumplir las consignas. Vale remarcar que la calidad de las abstracciones será un factor que puede influenciar a la nota del alumno.

#### 3.3. Contextualización

Se requiere del alumno modelar y construir el conjunto de clases que permitan la validación de movimientos de un juego de Ajedrez.

El sistema deberá permitir el ingreso de una cantidad arbitraria de piezas en un tablero estándar de 8 x 8. Para cada pieza deberá poder especificarse: el tipo de pieza (peón, caballo, alfil, torre, dama, o rey) y su posición en el tablero.

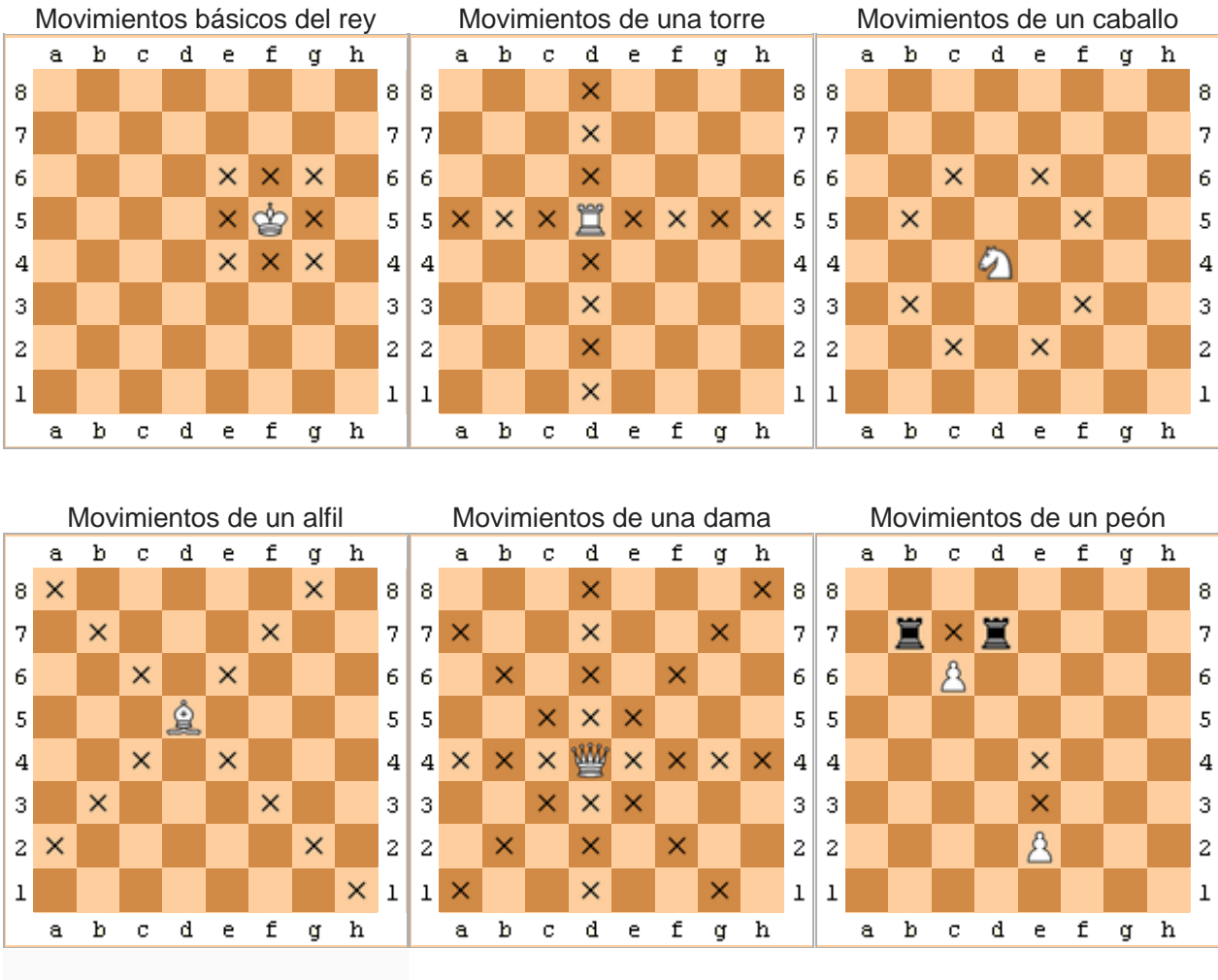
Una vez ingresadas todas las piezas deseadas, el sistema deberá calcular todos los movimientos posibles para cada pieza y presentar los mismos al usuario. Se recuerda al alumno que la notación algebraica del ajedrez define para cada casilla del tablero un nombre compuesto por su columna (letras de la "a" a la "h") y su fila (números del 1 al 8).



Consideraciones para tener en cuenta:

- Una pieza no puede ocupar un casillero ya ocupado por otra pieza de su mismo bando pero puede capturar una pieza del bando contrario.
- El rey no puede ser comido ni moverse a un casillero que lo deje en jaque.
- El enroque es un movimiento del rey y tiene 2 variantes: corto (flanco del rey) y largo (flanco de la dama).
- Los peones se mueven y capturan de formas diferentes. Asimismo, en su posición inicial, los peones pueden moverse 2 casilleros hacia adelante en lugar de sólo uno.

Se reproducen a continuación los movimientos básicos de todas las piezas.



3.4. Consignas	
1. Implementar el cálculo de movimientos para el alfil, la torre, y la dama blancos.	
2. Implementar el cálculo de movimientos para el alfil, la torre, y la dama negros.	
3. Implementar el cálculo de movimientos para el peón, el caballo, y el rey blancos.	
4. Implementar el cálculo de movimientos para el peón, el caballo, y el rey negros.	
5. Implementar el cálculo del enroque (en sus 2 variantes)	
6. Para cada pieza, ordenar sus posibles movimientos de mejor a peor considerando sólo la ganancia de material. Usar los siguientes valores de piezas: <ul style="list-style-type: none"><li>- Dama: 9 puntos</li><li>- Torre: 5 puntos</li><li>- Alfil/Caballo: 3 puntos</li><li>- Peón: 1 punto</li></ul>	