

Ejercicios Resueltos Diagrama de Clase y de Secuencia

El **tablero de damas** está compuesto de 64 **celdas** en una grilla de 8x8. El objetivo del juego es *capturar* todas las **piezas** de tu oponente. Los **contendientes** usan su turno para *mover* una de sus piezas del tablero a una celda vacía. Una **movida válida** es una movida diagonal hacia adelante en una celda hacia una vacía o una movida hacia delante saltando sobre una pieza del oponente hasta una celda vacía. La última movida captura la pieza saltada. La pieza saltada es *removida* del tablero.

Después de capturar la pieza del oponente, se puede repetir el mismo procedimiento de captura mientras sea posible capturar otra pieza.

Si un **jugador** logra mover una de sus piezas hasta el borde del tablero del oponente, entonces esta pieza es *promovida* a **Dama**. Una Dama tiene poderes especiales: además de poder moverse diagonalmente hacia adelante, puede también hacerlo hacia atrás.

El juego se gana cuando el jugador ha capturado todas las piezas de su oponente (se puede empatar si un jugador no puede moverse más o ninguno puede comer piezas del otro).

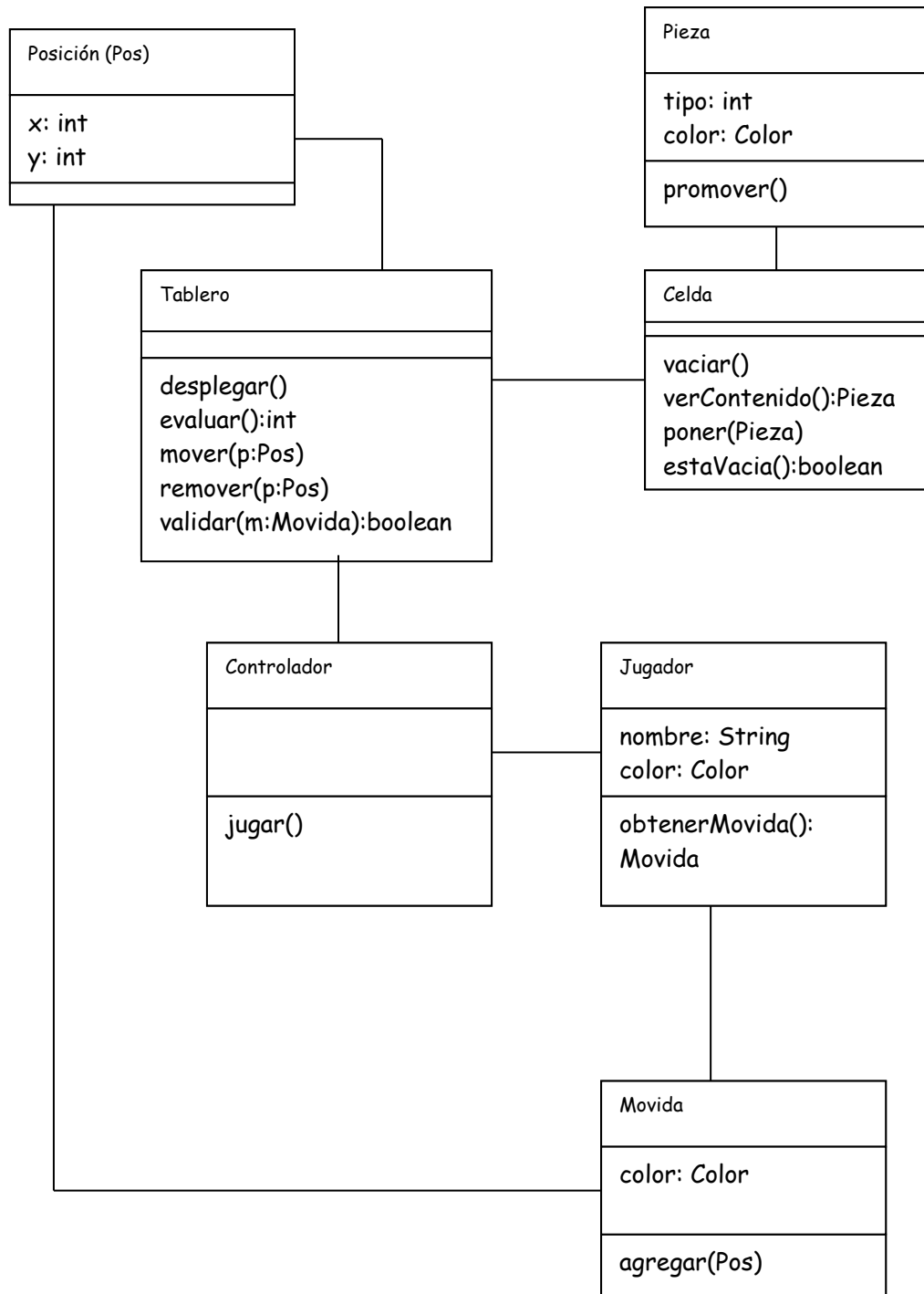
Dado que el juego ocurre en una simulación en el computador, se necesitará un **controlador** del juego que le *pregunta* a cada jugador de turno su movida. Cuando se reciba una movida de un jugador, se solicita al tablero que valide la movida. Si ésta es válida entonces la pieza del jugador actual es movida en el tablero. Si se captura alguna pieza ésta es removida del tablero. El tablero es *desplegado* y se *evalúa* el nuevo estado del juego. Este proceso se repite hasta que un jugador gane al dejar a su oponente sin piezas o se llegue a una posición de bloqueo.

1. Especificar el diagrama de clases que represente este juego.
2. Dibujar el diagrama de secuencia para la inicialización y la primera jugada, en base a los siguientes segmentos de código de la implementación.

```
class Controlador
// ...
T.inicializar()
Movida m = Jturno.obtenerMovida() ;
boolean movOK = T.validar(m) ;
if (movOK) {
    T.ejecutar(m) ;
    cambiarTurno() ;
}
// ...

class Jugador {
...
public Movida obtenerMovida() {
    Movida m = new Movida() ;
    ...
    return m ;
}
...
}
```

1. Posible SOLUCION Especificar el diagrama de clases que represente este juego.



2. POSIBLE SOLUCIÓN Dibujar el diagrama de secuencia para la inicialización y la primera jugada, en base a los siguientes segmentos de código de la implementación.

```
class Controlador
// ...
T.inicializar()
Movida m = Jturno.obtenerMovida() ;
boolean movOK = T.validar(m) ;
if (movOK) {
    T.ejecutar(m) ;
    cambiarTurno() ;
}
// ...

class Jugador {
...
public Movida obtenerMovida() {
    Movida m = new Movida() ;
    ...
    return m ;
}
...
}
```

