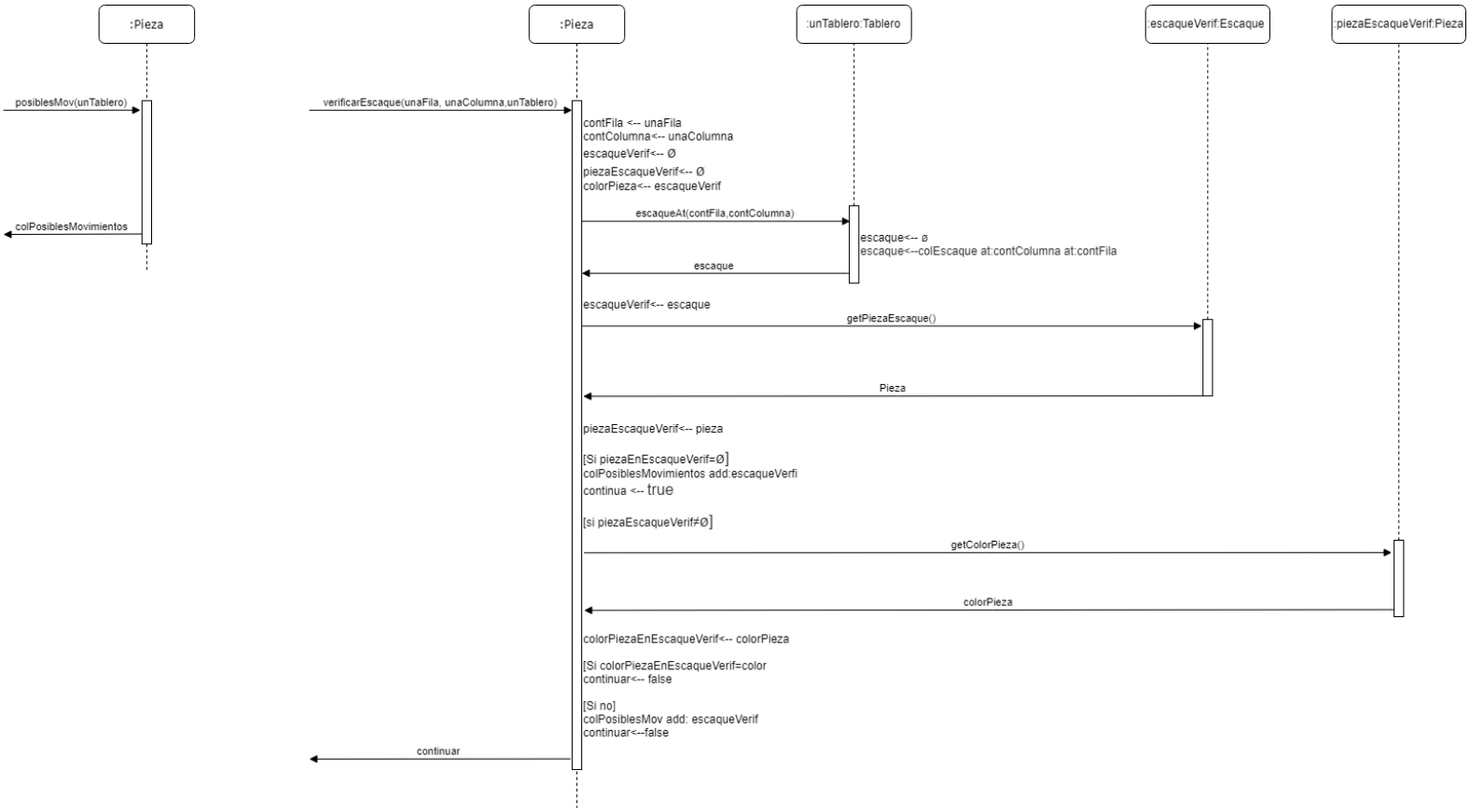
DIAGRAMAS E INFORME AJEDREZ

Manuel Agustin Latorre Rosales

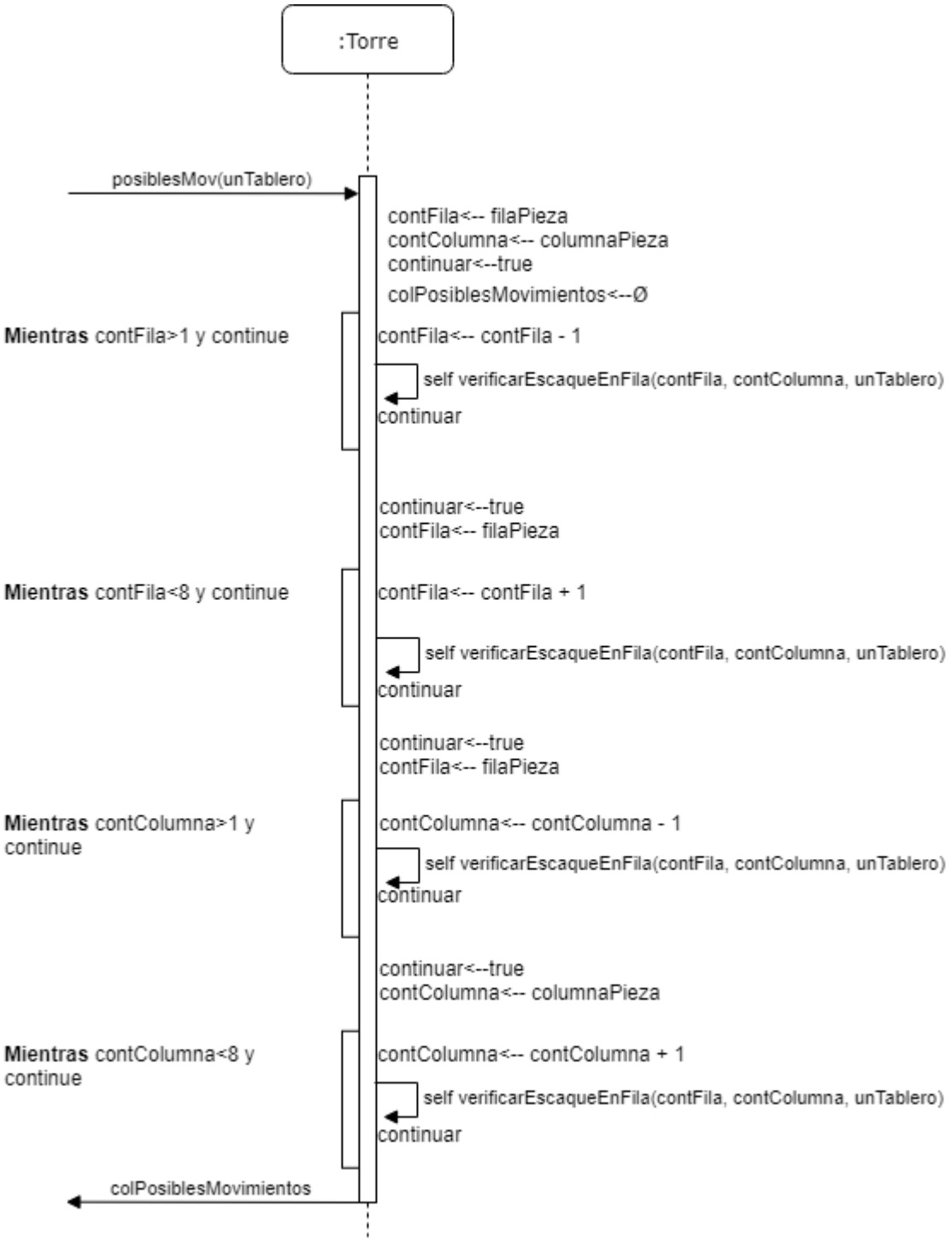
POO TRABAJO OBLIGATORIO LIBRE 2019

Diagrama de clases Ajedrez

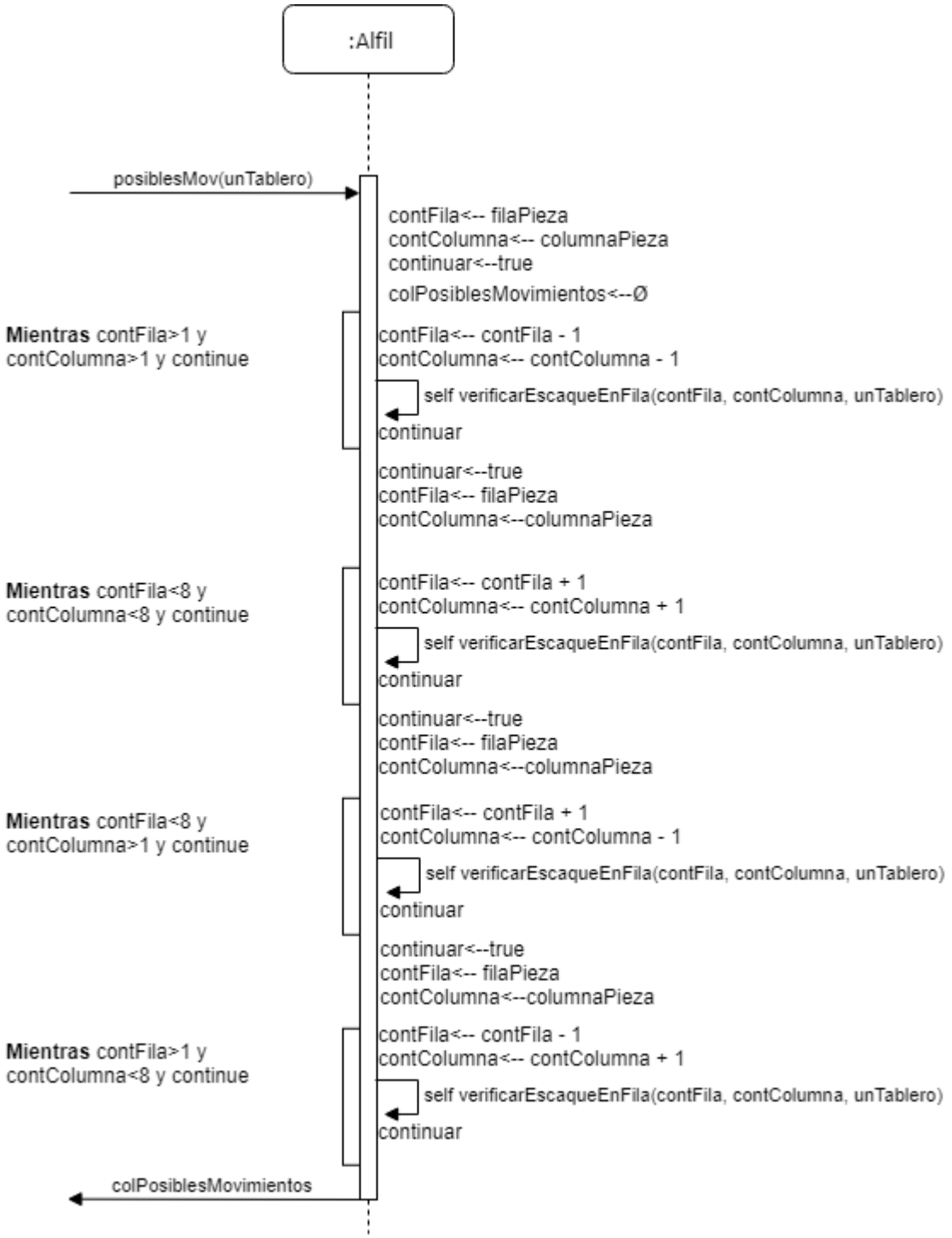
Diagrama de Secuencia mensaje posibles movimientos y método verificarEscaque



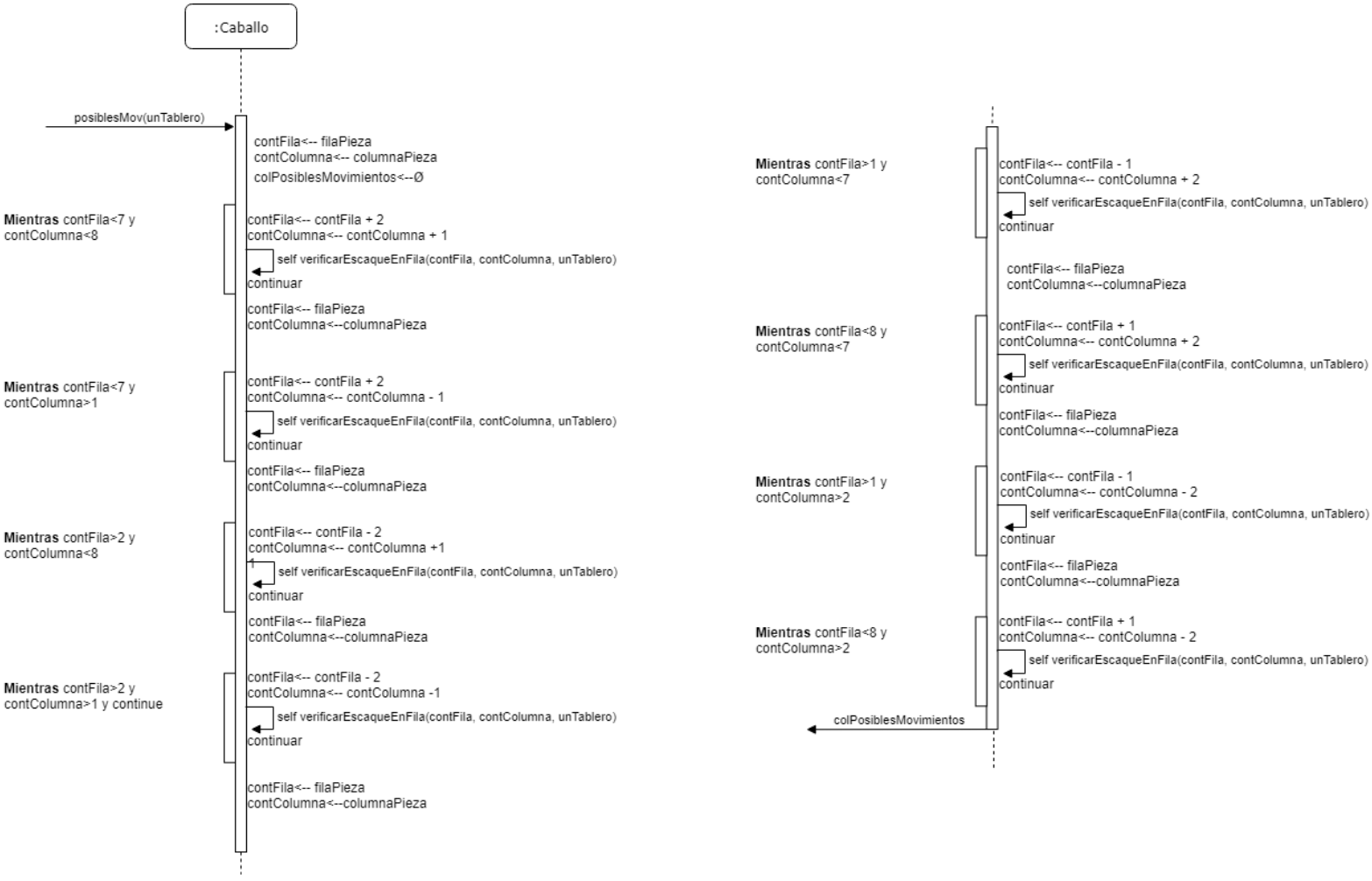
Mensaje polimórfico a Torre



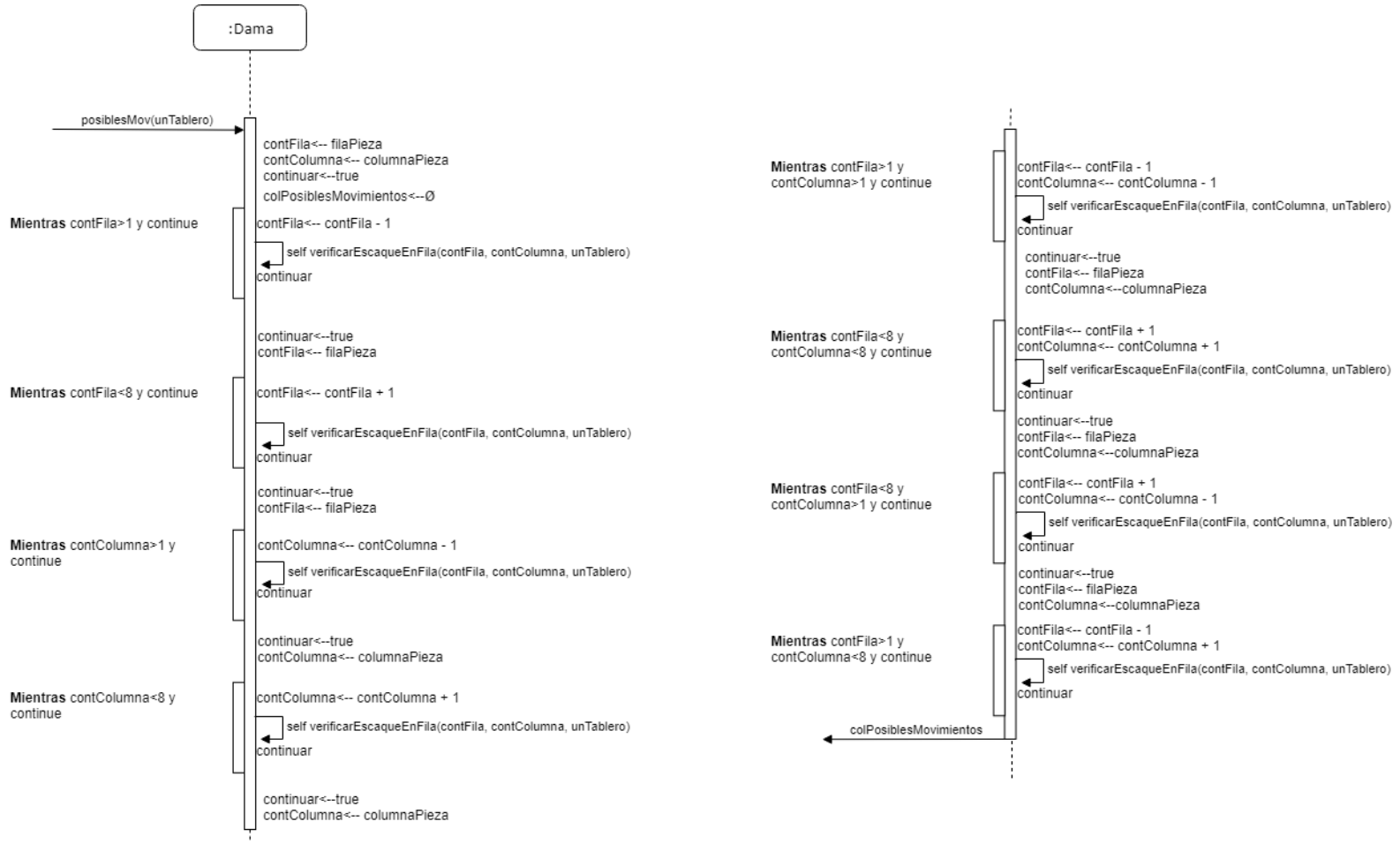
Mensaje polimórfico a Alfil



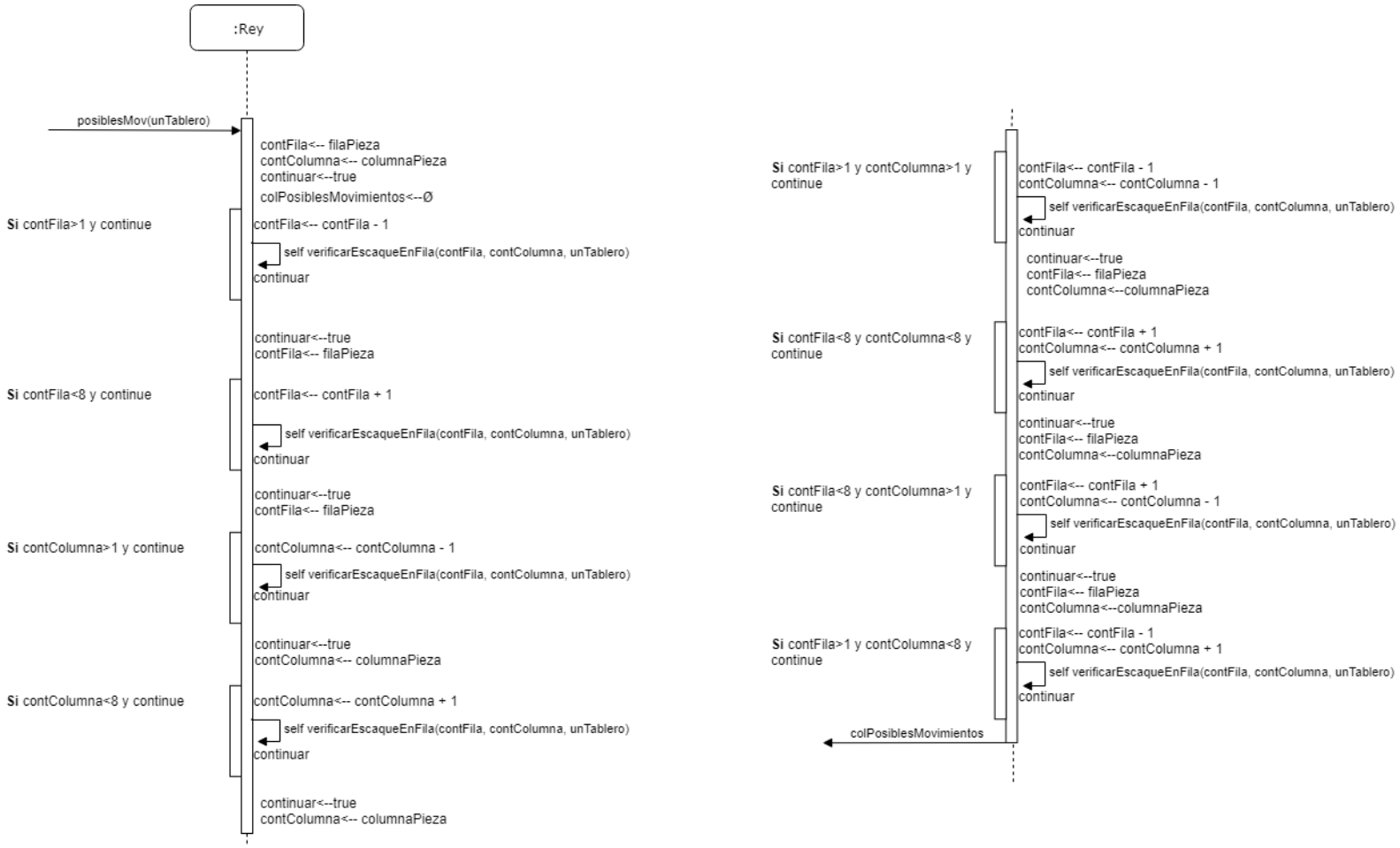
Mensaje polimórfico a Caballo



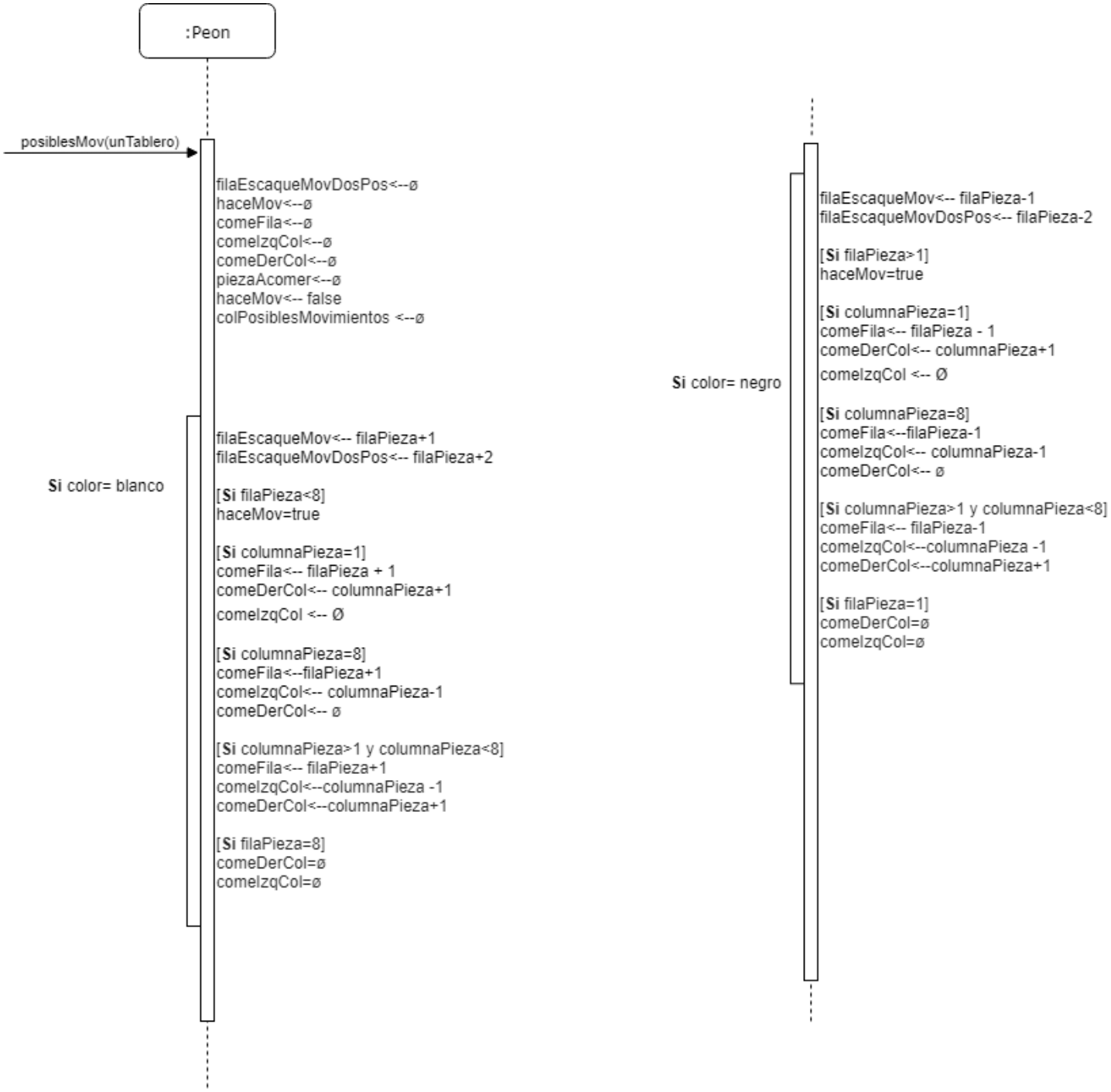
Mensaje polimórfico a Dama



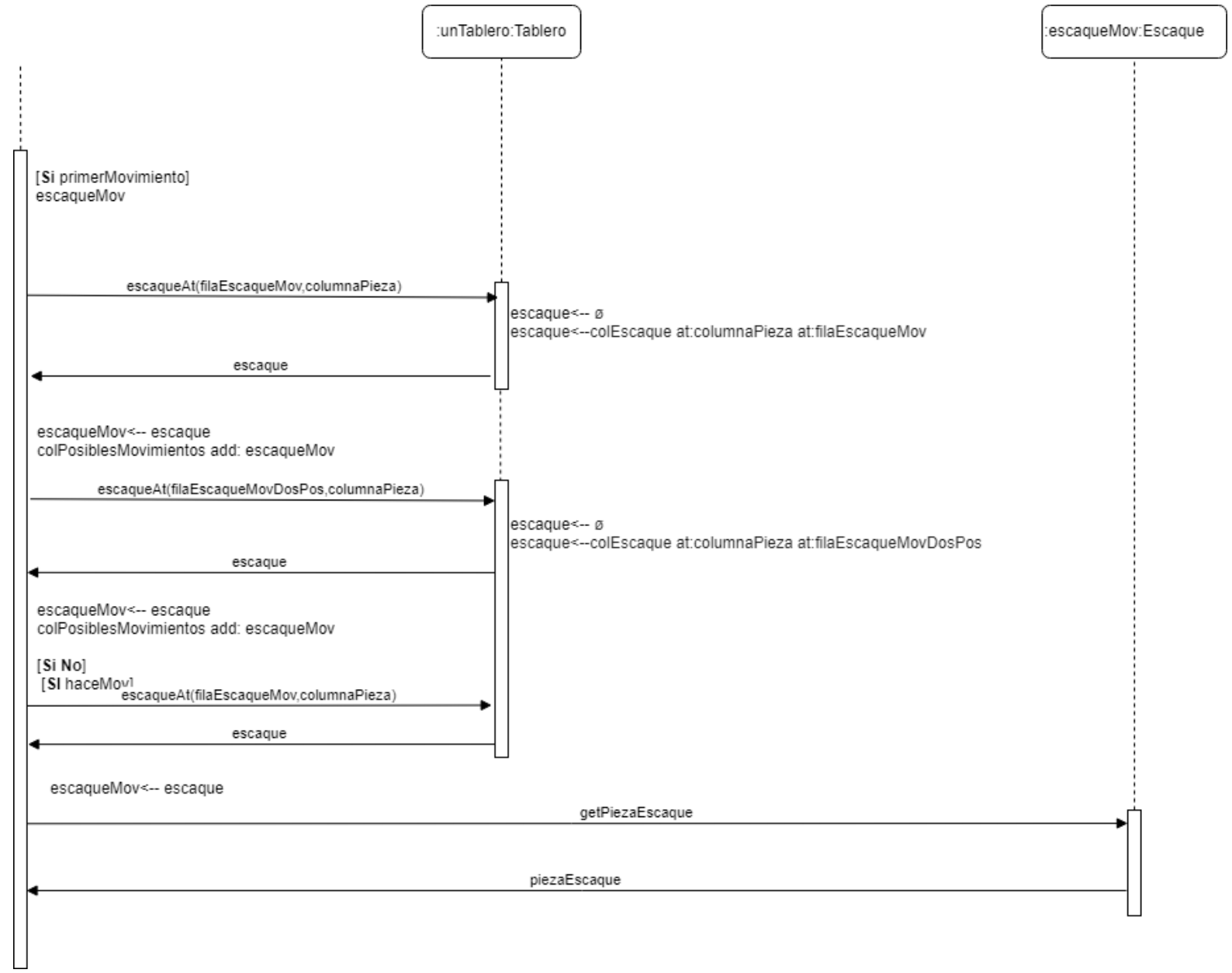
Mensaje polimórfico a rey



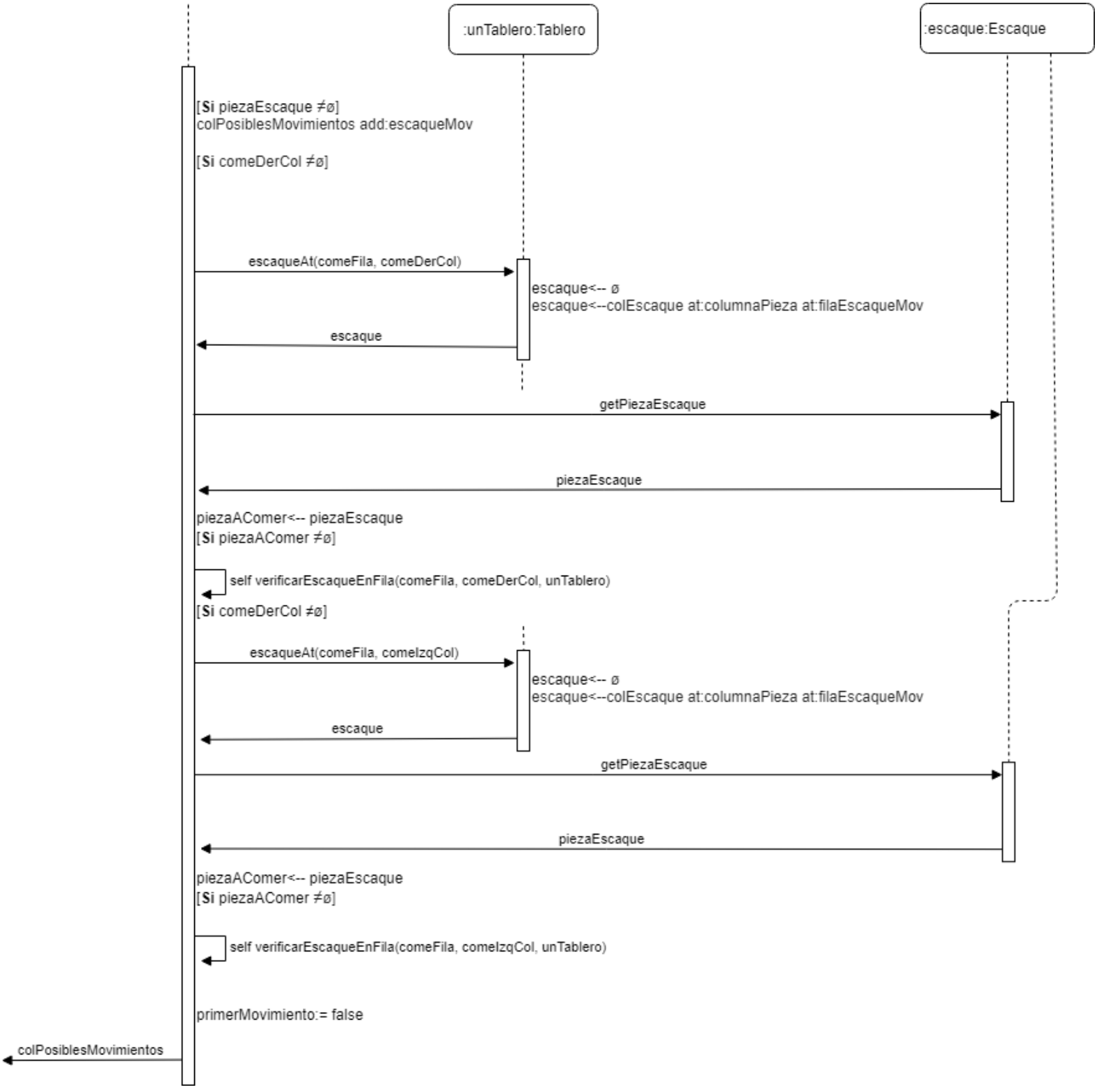
Mensaje polimórfico a Peon (Parte 1)



Mensaje polimórfico a Peon (Parte 2)



Mensaje polimórfico a Peon (Parte 3)



Modelo del juego (clases del dominio)

Investigué sobre diferencias entre diagrama de clases y modelo UML (ya que en del apunte no me quedaba muy claro) y la única diferencia que encontré es que además de los atributos el modelo UML muestra los métodos de cada clase. Por lo que opte por dejar las mismas relaciones y multiplicidades que en el diagrama de clases.

INFORME

Reglas del juego e indicaciones:

- No se juega con “enroque”.
- No se juega con “jaque mate”.
- No se juega con “coronar al peón”, esto es cuando el peón llega al borde contrario

El juego simplificado puede terminar por tres razones:

1. Se completaron las n jugadas preprogramadas
2. Decisión de los jugadores
3. Se “derrocó” al rey del otro jugador.

Ya que no se juega con “jaque mate” y el juego termina “derrocando” al rey no implemente un método que controle si un movimiento, propio o de otra pieza, deje al rey en jaque.

El tablero tiene una sola perspectiva (piezas negras abajo, piezas blancas arriba) es decir no cambia dependiendo del turno del jugador por lo que las filas y columnas van a ser las mismas para ambos jugadores.

Para que las piezas se carguen adecuadamente la carpeta ‘PiezasAjedrez’ debe estar en la carpeta de la imagen de pharo que se esté ejecutando (**NO CAMBIAR EL NOMBRE DE LA CARPETA**)

En caso de que no carguen las imágenes se deberá reemplazar manualmente la dirección de estas por donde las tenga guardadas en:

Partida>> imagenCorrespondiente (por ejemplo, **Partida>> alfilBlanco** ingresar la dirección de la imagen entre los apostrofes. Asi con cada una de las imagenes)

Y en **Tablero>> bordes** (las direcciónes que estan después de Form fromFileNamed).

Implementación a Smalltalk:

Los objectos Tablero, Piezas y Jugadores se crean en la clase Partida (es creada en el playground con el mensaje Partida new) cuando se selecciona la opción empezar partida.

Los escaques son creados por el tablero mismo ya que este es una matriz que se forma a partir de estos, para esto me base en el ejemplo del juego LightsOut del libro PharoByExample.

Tanto las piezas como los jugadores son creados de manera individual ya que cada uno es representado como un objecto único (No se me ocurrió una manera mejor de hacerlo para luego trabajar con ellos), cada pieza y cada jugador tiene asignado su color correspondiente.

Las imágenes de cada pieza las ‘llamo’ en un método único para cada una de manera de que en caso de que se quiera o deba modificar la dirección donde se encuentran guardadas sean de fácil acceso, en Windows si la carpeta se encuentra en la carpeta de la imagen de pharo en la que se esta trabajando con la dirección ‘PiezasAjedrez\pieza.png’ la carga sin problemas, en Linux ya necesita toda la ruta desde home\ completa.

El mensaje polimórfico que utilice para el mensaje PosiblesMovimientos de las piezas es un polimormismo por reemplazo, no se me ocurrió uno mejor ya que, si bien el código es muy similar y la idea es la misma (recorro en las posibles direcciones de cada pieza) este es distinto

en cada caso ya que los recorridos de estas son distintos para cada pieza. Para la dama y el rey se me ocurrió usar los movimientos (métodos) de Torre y Alfil para obtener sus movimientos, pero estos se obtienen a travez del mensaje polimórfico de Pieza y según lo que investigue no se puede, entonces reutilice el código de torre y alfil para calcular los movimientos de la Dama y el Rey.

También utilice un método implementado en la clase pieza que me verifique si un escaque (recorrido desde el polimorfismo de la pieza) es valido para recibir la pieza, si es válido el método se encarga de agregarlo a la colección de posibles movimientos y devuelve si es posible seguir encontrando escaques en esa dirección (con esto controlo que no se salteen piezas).

El ingreso de columnas funciona tanto con letras (mayúsculas y minúsculas) como con números ya que la matriz Tablero se recorre con numero de fila y columna, para tansformar las letras ingresadas en números (y asi poder recorrer la matriz) implemente un metodo 'cambioCharAnumero:' en el que transformo cada una de las letras validas en su numero correspondiente, no logre encontrar un método que me simplifique esto

Problemas y dificultades que tuve:

No pude encontrar una manera de asignarle a cada jugador un cronometro.

No pude asignarle una perspectiva única a cada jugador del tablero ya que esto implicaría crear el tablero nuevo, invertir filas y columnas y reubicar las piezas cada vez que se cambie de turno por lo que opte por dejar una sola perspectiva para ambos jugadores

No realice primero los diagramas y luego pase a la implementación como veníamos trabajando en la cursada, ya que como no tuvimos practica con Morphs y UIManager no sabía que podía y que no podía hacer ni el cómo, por lo que fui programando a medida que aprendía apoyándome en diagramas de secuencia para problemas complejos como los posibles movimientos de las piezas, realice un diagrama de clases antes de arrancar a programar para poder tener una base y a medida que iba programando y se me ocurrían cosas nuevas o maneras de pensarlo mejor fui modificando en el diagrama de clases (las multiplicidades las mantuve y solo agregue una relación entre jugador y tablero, también agregue algunas variables de instancia y elimine otras que termine no utilizando)