**TP GRUPAL: SNAKE GAME**

**POLIMORFISMO**

* *¿Cuál es el mensaje polimórfico? ¿Qué objetos lo implementan? ¿Qué objeto se aprovecha de ello?*

El mensaje polimórfico es direccion(unaDireccion), el cual es implementado por los objetos normal y locura. El objeto que se aprovecha de ello es serpiente.

**COLECCIONES**

* *¿Qué operaciones de colecciones se utilizan? ¿Usaron mensajes con y sin efecto? ¿Para qué?*

Las operaciones de colecciones que se utilizan son forEach(bloqueConEfecto), last(), size(), remove(objeto), add(objeto), isEmpty() y filter(bloqueDeBool). Los mensajes isEmpty(), last() y size() no generan efecto mientras que el resto de los mencionados previamente lo hacen.

Los mensajes forEach(bloqueConEfecto), last(), size(), remove(objeto) y add(objeto) se implementaron en la lista cuerpo del objeto serpiente para agregar o sacar partes del cuerpo y asignarles anterior y siguiente. El mensaje isEmpty() se usó para saber si había algún elemento en una posición dada. Por último, filter(bloqueDeBool) se utilizó para filtrar posiciones que no estén ocupadas para asignarle alguna a las pociones.

**CLASES**

* *¿Usan clases? ¿Por qué? ¿Dónde o cuándo se instancian los objetos?*

Utilizamos clases para Muro, Nivel, CuerpoSnake, ObjetoAleatorio y Pocion porque necesitábamos objetos con diferentes identidades que tuvieran comportamiento y atributos en común.

Los objetos de la clase Muro se instancian cuando se inicia el nivel, mientras que los objetos de la clase CuerpoSnake se instancian cuando se agrega una parte al cuerpo de la serpiente. Los objetos de las demás clases se instancian al momento de declararlas, tales como cabezaSnake, nido y nivel1.

**HERENCIA**

* *¿Entre quiénes y por qué? ¿Qué comportamiento es común y cuál distinto?*

Hay herencia entre ObjetoAleatorio y Pocion porque Pocion posee el mismo comportamiento y los mismos atributos que ObjetoAleatorio, pero la anterior requería más métodos.

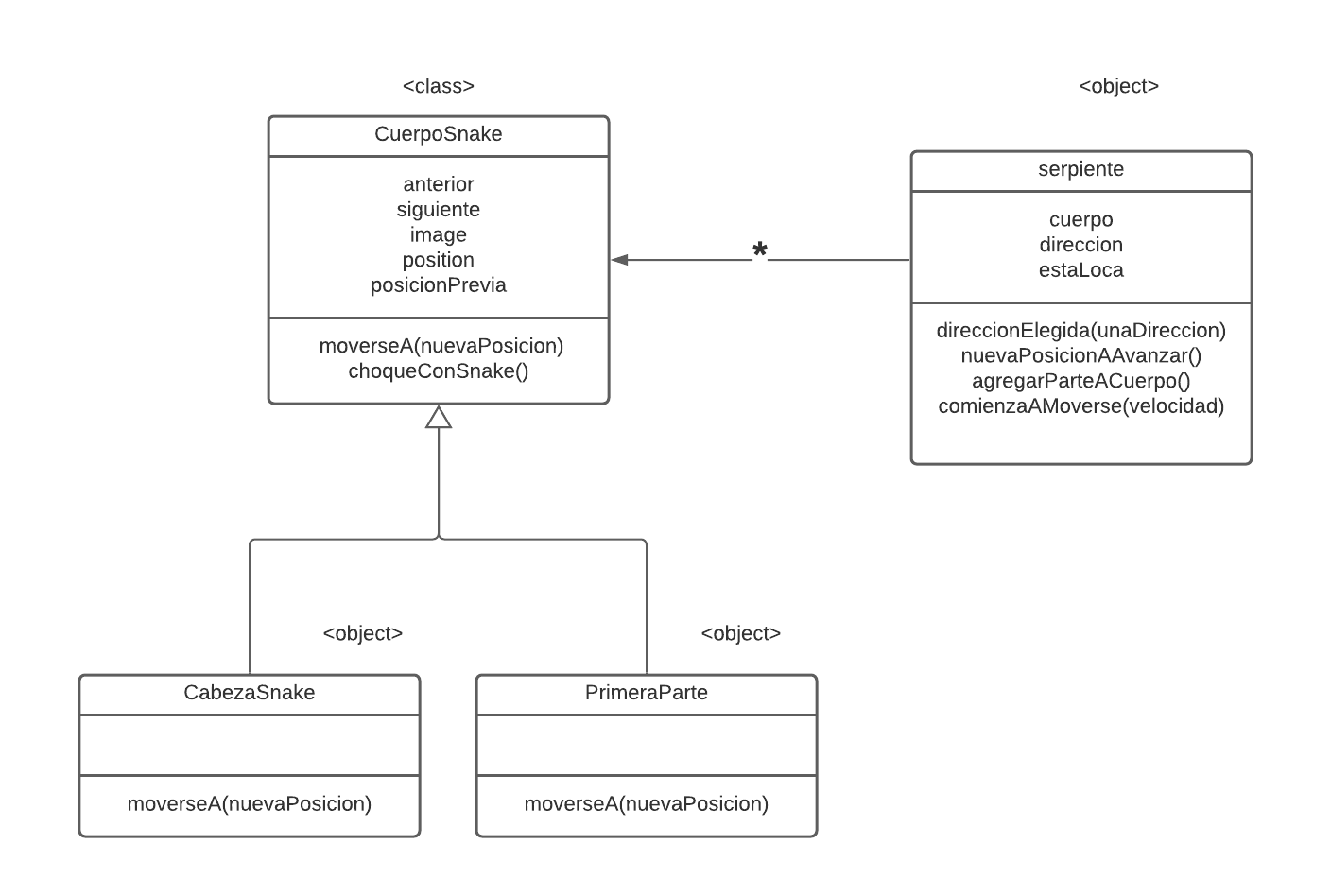
El comportamiento en común son el método puedoIrEn(posicionTentativa) y posicionAleatoria(limite) mientras que los distintos son volverANormalidad(), posicionTentativa(unaDireccion), choqueConSnake() y volverANormalidad(), entre otros.

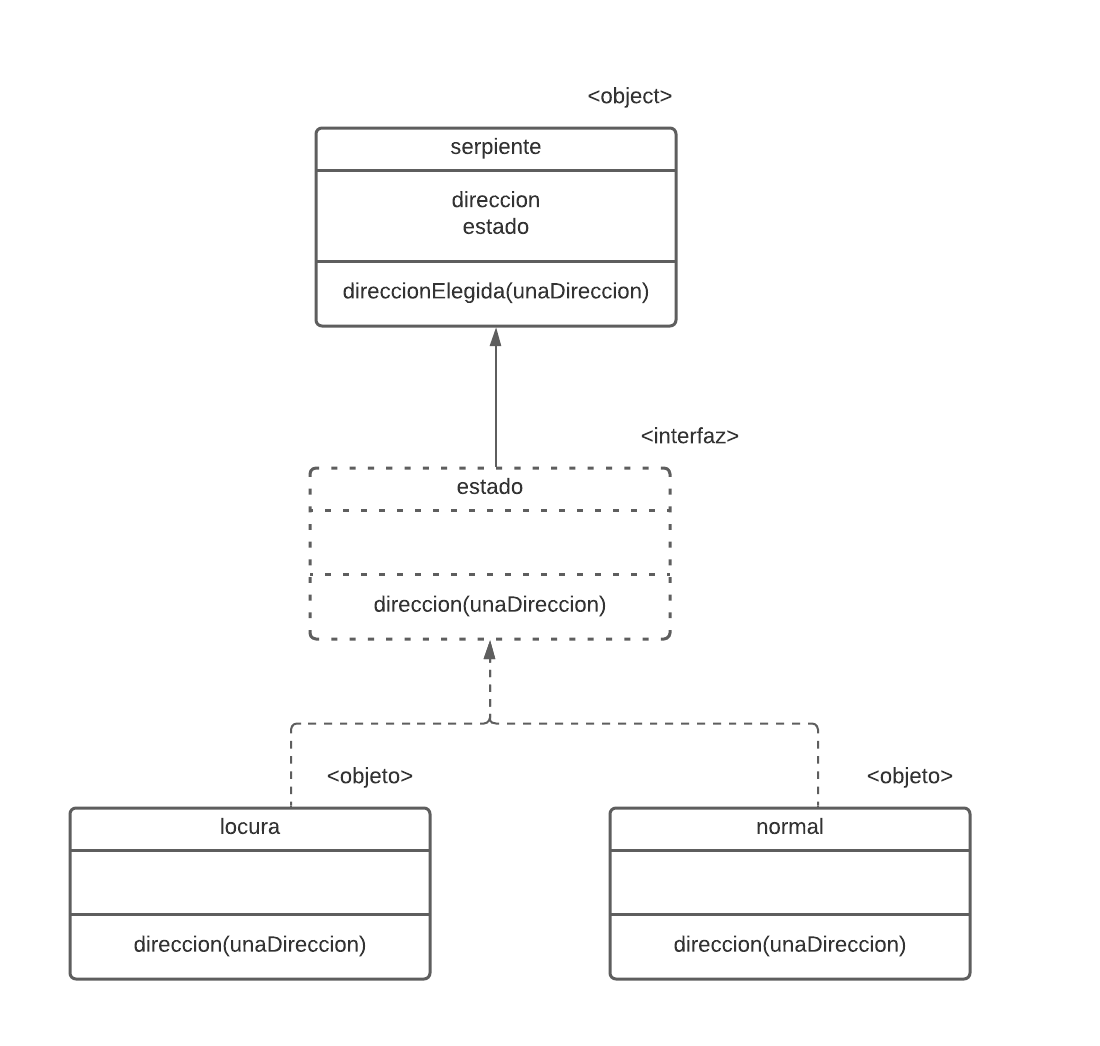
**COMPOSICIÓN**

* *¿Qué objetos interactúan? ¿Dónde se delega? ¿Por qué no herencia?*

Interactúan el objeto serpiente con sus respectivos estados, que pueden ser locura o normal. Se aplica este concepto cuando tenemos que definir el estado de la serpiente construyendo un objeto para cada estado, y dentro de los objetos el mensaje polimórfico direccion(unaDireccion), que dependiendo el estado devuelve la dirección a seguir. No usamos herencia ya que no permite el cambio de estado, a diferencia de la composición.

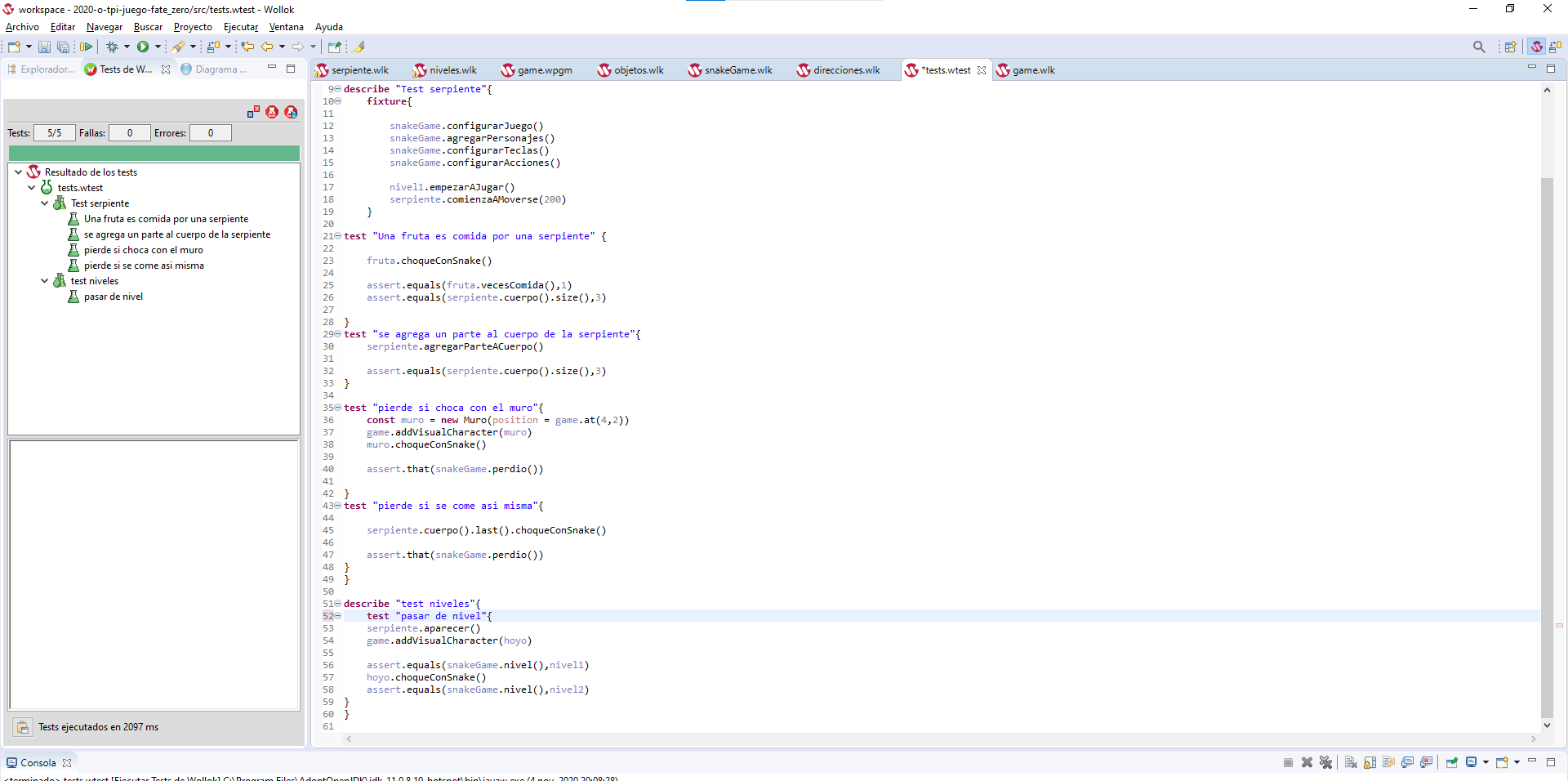
**DIAGRAMA**

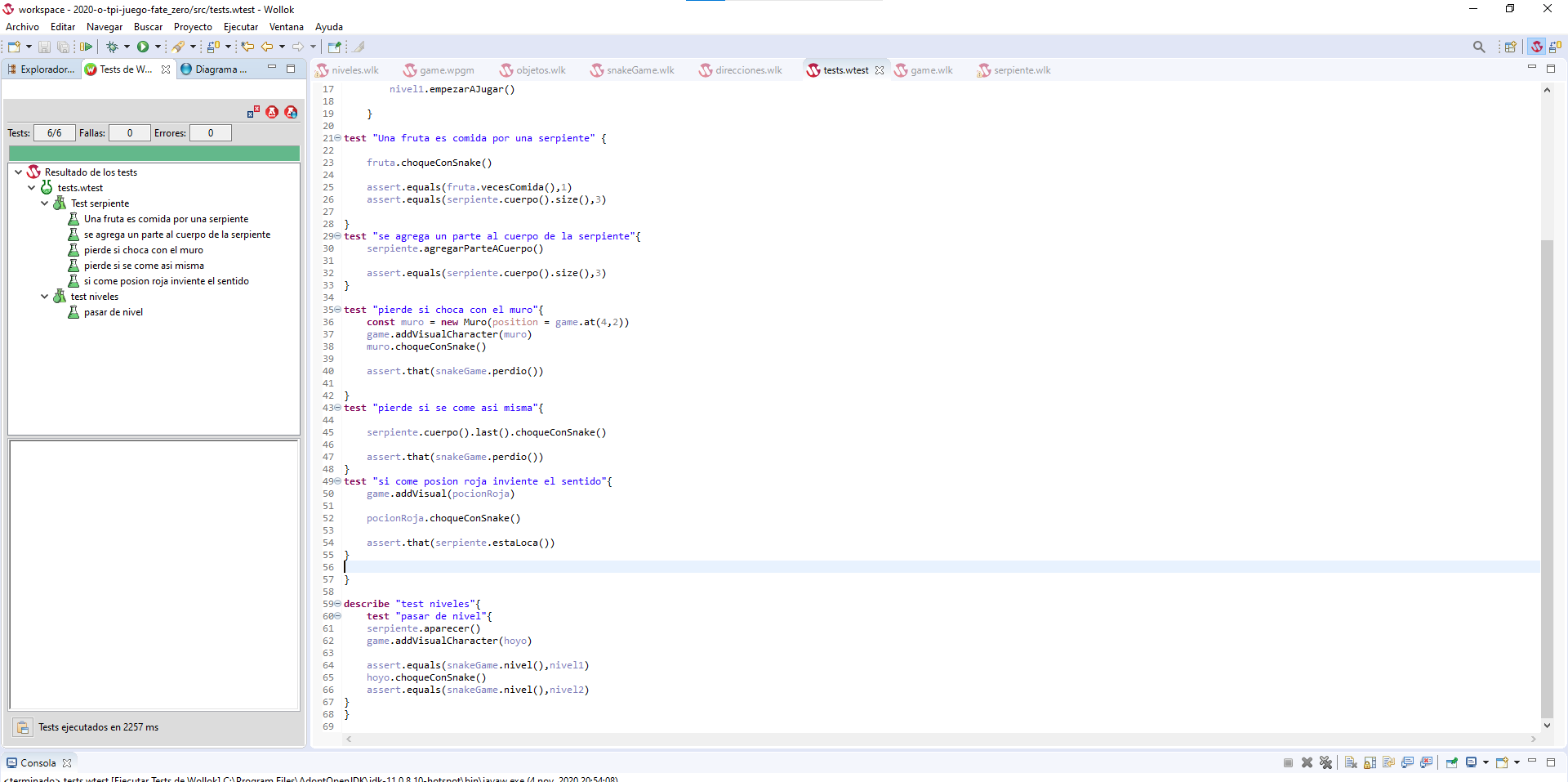
****

****

**Versión 2:** con composición de estados

**Ejecutando tests (sin sound)**

****

****