**FIUBA – 75.07**

**Algoritmos y programación III**

*Trabajo practico 2: Dragon AlgoBall*

1er cuatrimestre, 2017

(trabajo grupal de 4 integrantes)

Alumnos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Padrón** | **Mail** |
| Axel Erlich | 94847 | Axel.erlich@yahoo.com.ar |
| Patricio Pizzini | 97524 | pizzinipatricio@yahoo.com.ar |
|  |  |  |
|  |  |  |

***Fecha de entrega final***: 22 de junio

***Tutor***: Tomas Bustamente

***Comentarios***:

**Informe**

**Supuestos**

El personaje va a poder pasar por encima de otro, pero no ocupar la misma celda.

La vida de cell puede aumentar mas alla de lo establecido al comienzo.

**Modelo de dominio**

El proyecto cuenta con varios paquetes:

Excepciones, juego, personajes y tablero.

Dentro del paquete excepciones se encuentran declaradas las excepciones que se manejan durante el desarrollo del código que son explicadas en el ultimo apartado del informe actual.

En el paquete juego, se declaran las clases Juego que es donde se desarrolla el comienzo y fin del mismo, los turnos, asignaciones de equipo, movimientos de los personajes.

Jugador que representa a cada jugador que tomara parte en el juego.

JugadorEquipoVillano y JugadorEquipoZ, ambas heredan de JugadorEquipo y las tres implementan sus respectivas interfaces, básicamente se les asigna a cada jugador los 3 persones respectivos.

Ademas cuenta con la clase Mapa.

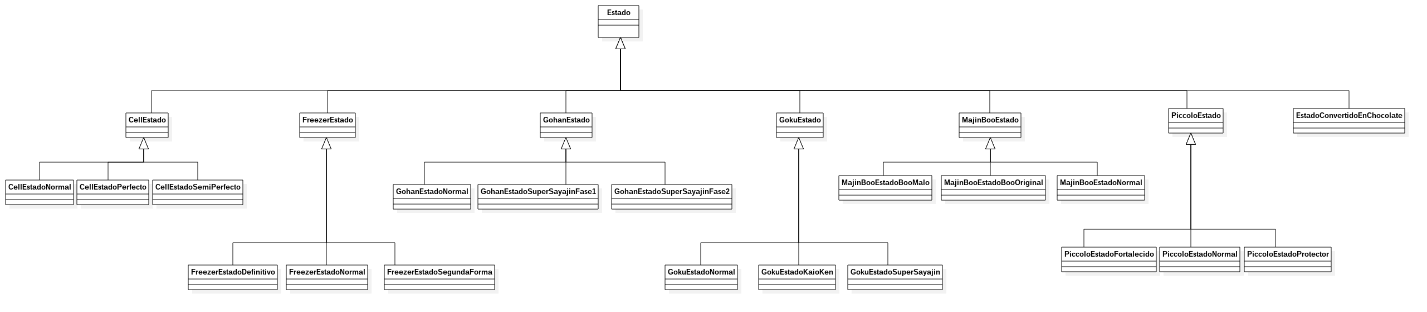
Asi también tenemos el paquete personajes, donde como principal característica tenemos la clase Personaje, la cual es madre de los 6 personajes del juego. Cada uno de estos personajes implementa una interfaz IPersonajeEquipoXXXX siendo XXXX villano o Z dependiendo que tipo de personaje es cada uno.

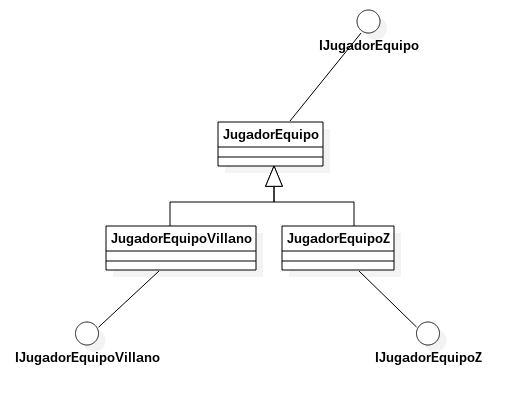
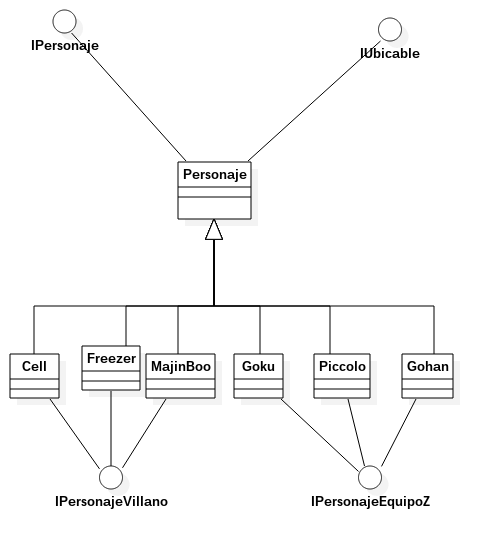
Ademas cada personaje cuenta con diferentes estados, que representa el estado propiamente dicho de cada personaje, tanto antes como después de una transformación.

Es decir, cada personaje tiene una clase XXXXEstado, que es clase madre y de ella heredan tres clases que representa cada transformación que puede llevar a cabo el personaje.

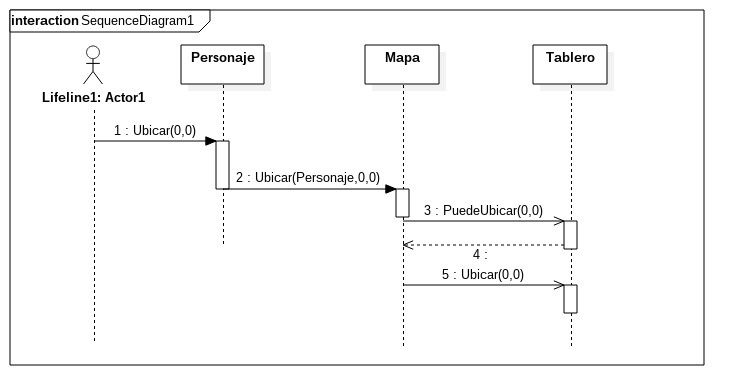
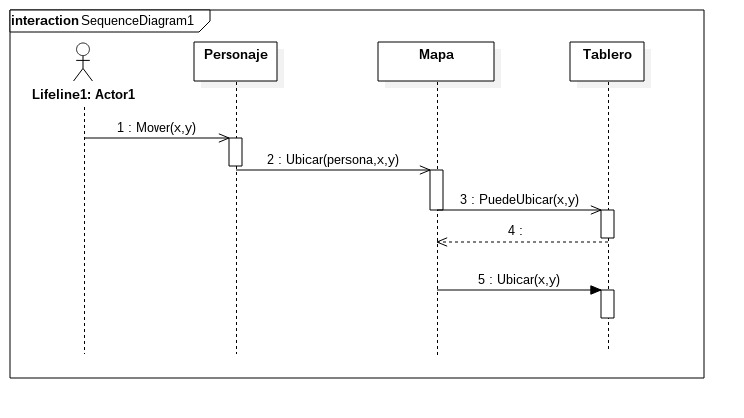
Finalmente, en el paquete del tablero, se encuentran las celdas, coordenadas, el tablero propiamente dicho, consumibles (clase madre) que pueden ser EsferaDelDrago, NubeVoladora y SemillaDelHermanito (clases hijas) y una interfaz IUbicable que es la encargada de ubicar a cada personaje y/o consumible.

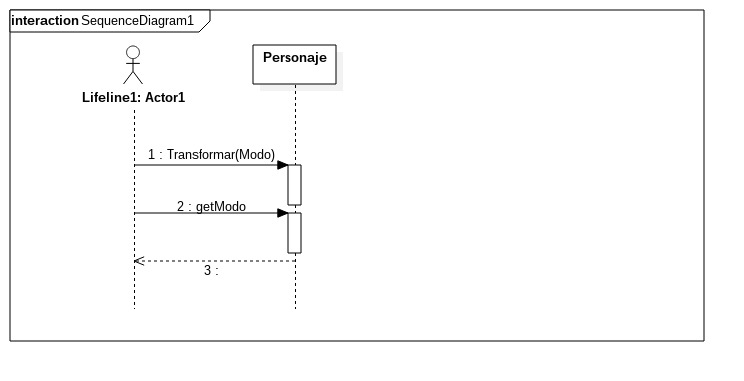
**Diagrama de clases**

****

****

**Diagramas de secuencia**





**Excepciones**

Actualmente, el proyecto cuenta con 21 excepciones, las cuales son:

\*AtaqueFueraDeRangoException: en cada ataque se verifica que el personaje que lo hara puede realizarlo, es decir que el personaje que recibirá el ataque se encuentre dentro de la distancica de ataque máxima.

\*AtaqueMismoEquipoException: se atrapa esta excepción cuando un personaje quiere atacar a otro de su mismo equipo.

\*CeldaNoExisteException

\*CeldaOcupadaException: como se dijo en el apartado de supuestos, un personaje no puede ser ubicado en la misma coordenada que otro personaje, cuando esto ocurra se atrapara la excepción.

\*HayJugadoresSinEquipoException

\*JugadoresNoSeleccionadosException

\*JugadorNoPoseeEquipoException

\*JugadorPoseeEquipoException

\*JugadorYaExisteException

\*MovimientoFueraRangoException

\*NoExisteElPersonajeException

\*NoPuedeRealizarElAtaqueException

\*NoPuedeTransformarException

\*PersonajeEquipoContrarioException

\*PersonnajeFueraDeCombateException

\*PersonajeInhabilitadoException

\*PersonajeNoExisteException

\*PersonajeYaEsChotolateException

\*UltimoEstadoException

\*YaRealizoAtaqueException

\*YaRealizoMovimientoException