FIUBA - 75.07

Algoritmos y programación III

Trabajo práctico 2: Dragon AlgoBall

1er cuatrimestre, 2017

(trabajo grupal de 4 integrantes)

Alumnos:

Nombre	Padrón	Mail
Federico Elias	96105	fedelias93@gmail.com
Pavlo Holota	99760	pavlo.holota@gmail.com
Julieta Caceres	96454	julieta.agustina.caceres@gmail.com
Jimena Roselló	98942	jimena.rosello@gmail.com

Fecha de entrega final: 22 de junio

Tutor:

Comentarios:

Informe

Enlace para editar:

https://docs.google.com/document/d/1ts8QLEb9fbDGD956ZlhpqHQFnPv5ma5x0pZQEqc__kQ/edit?usp=sharing

Supuestos

1. Los personajes de un mismo equipo aparecerán en diagonal de forma contigua, en esquinas opuestas del tablero.
2. El jugador no elige por qué casilleros puntuales se mueve el personaje.
3. Un consumible se absorbe cuando un personaje realiza un movimiento terminando en el casillero que lo contenga (no puede consumirlo simplemente pasando sobre él).
4. La cantidad de consumibles es fija por cada turno.
5. Los personajes se pueden mover en diagonal.
6. Los personajes pueden pasar a su última evolución sin necesidad de pasar por el primer cambio de estado, y lo hacen automáticamente cuando cumplen las condiciones necesarias.

Modelo de dominio

Para la realización de la primera etapa del trabajo práctico consideramos importantes las siguientes clases:

Juego: la responsabilidad de esta clase es coordinar las acciones que los usuarios del juego quieren realizar con el desarrollo interno del programa.

Partida: es la entidad del programa encargada de preparar la estructura donde los usuarios podrán interactuar con los personajes y de actualizar los estados del juego.

Equipo: cuando un jugador elige un equipo instancia un objeto de esta entidad, la cual define los personajes que el usuario puede usar y que no podrán ser cambiados por otros personajes, de manera tal que los personajes de un mismo equipo no se puedan atacar entre si. Equipo es una Interfaz madre.

GuerrerosZ: es una entidad hija de la entidad Equipo que posee a los personajes Goku, Gohan, Piccolo.

EnemigosDeLaTierra: es una entidad hija de la entidad Equipo que posee a los personajes

VerificadorDeMovimiento: es una entidad que recibe la acción de mover un personaje, verifica que los casilleros dados estén vacíos y en caso de no estarlo lanza una excepción.

Personaje: entidad con la que el usuario juega, cada uno tiene su estado inicial con determinadas características de juego, pudiendo cambiar esta siempre que se cumplan las condiciones necesarias; pertenece a un equipo.

Asimismo, existen seis clases hijas de Personaje, que implementan la interfaz correspondiente a su equipo: Goku, Gohan, Piccolo, Cell, Freezer y MajinBoo.

Interfaces de Estado: son seis, una por personaje, y contienen los comportamientos de juego de un personaje en particular, cada una en un momento determinado.

El estado del personaje es el encargado de atacar, recibir ataques, cambiar las coordenadas del personaje, etc.

Clases de Estado: implementan las interfaces, y hay una por cada posible estado de cada personaje (estado normal, primera evolución, segunda evolución, etc.).

Consumibles: es una interfaz que representa un elemento que puede ser absorbido por un personaje para brindarle propiedades.

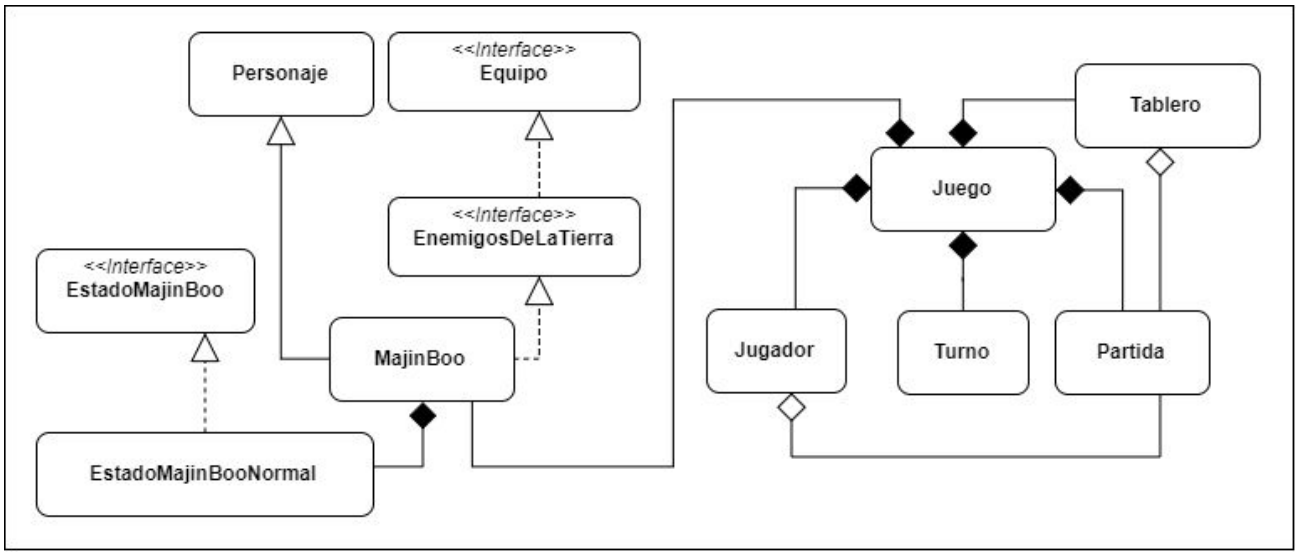
Es implementada por las clases **EsferaDelDragon**, **NubeVoladora** y **SemillaDelErmitanio**.

Casillero: objeto que es parte del tablero, este podrá estar vacío o contener un consumible o un personaje.

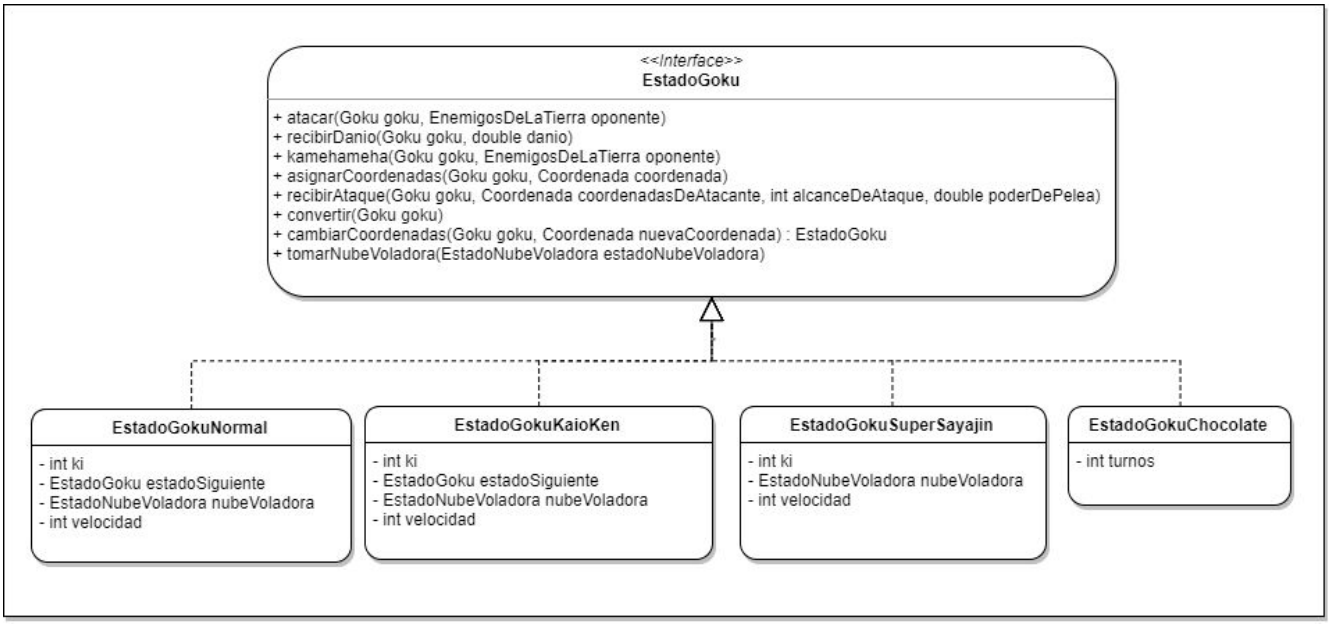
Tablero: Es el objeto que contendrá a los casilleros.

Diagramas de clases

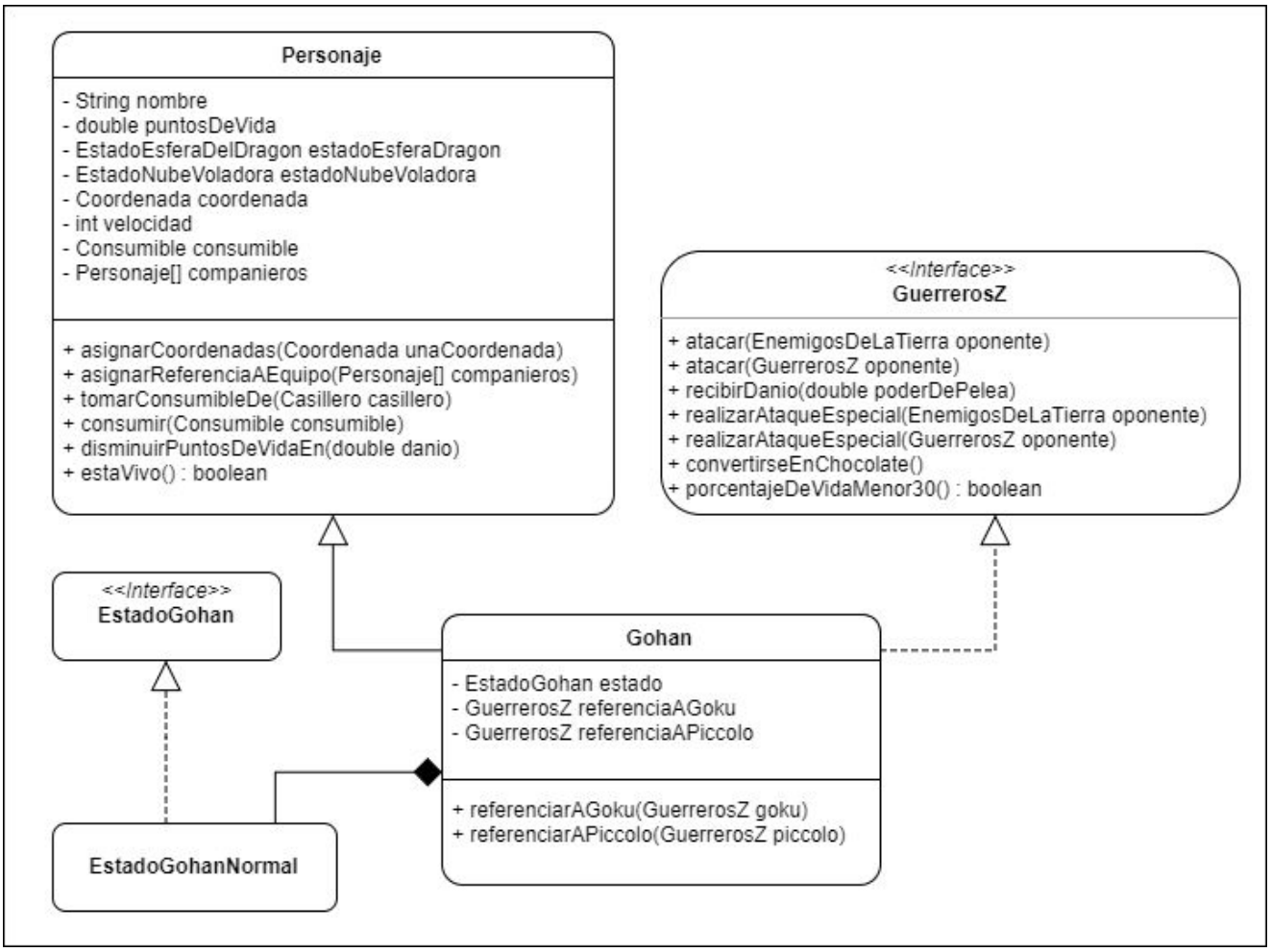
En primer lugar exponemos un diagrama de clases global de la aplicación, con la estructura de clases e interfaces del modelo pensado, y las relaciones entre ellas:



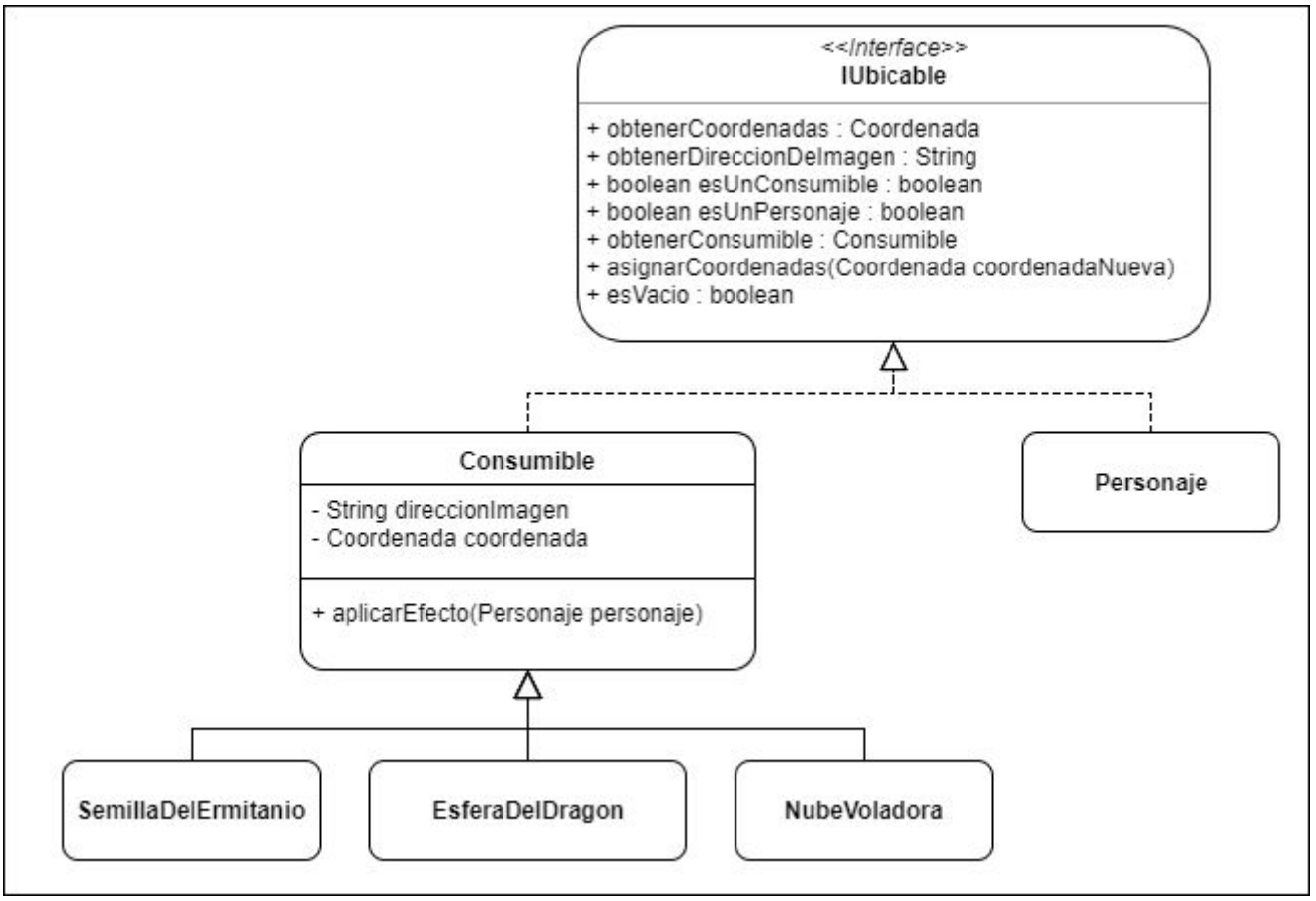
A continuación mostramos también un diagrama representando los Estados de Goku. Incluimos la interfaz EstadoGoku y las clases que implementan sus métodos. El diagrama es análogo para los demás personajes, con la única diferencia de que Cell, Freezer y Majin Boo no tienen un Estado de chocolate, ya que no pueden ser convertidos en chocolate.



Por otro lado realizamos un diagrama de clases sobre Gohan y las clases e interfaces con las que interactúa. El diagrama es análogo para los demás personajes, con la diferencia de que Cell, Freezer y Majin Boo implementan EnemigosDeLaTierra en lugar de GuerrerosZ.



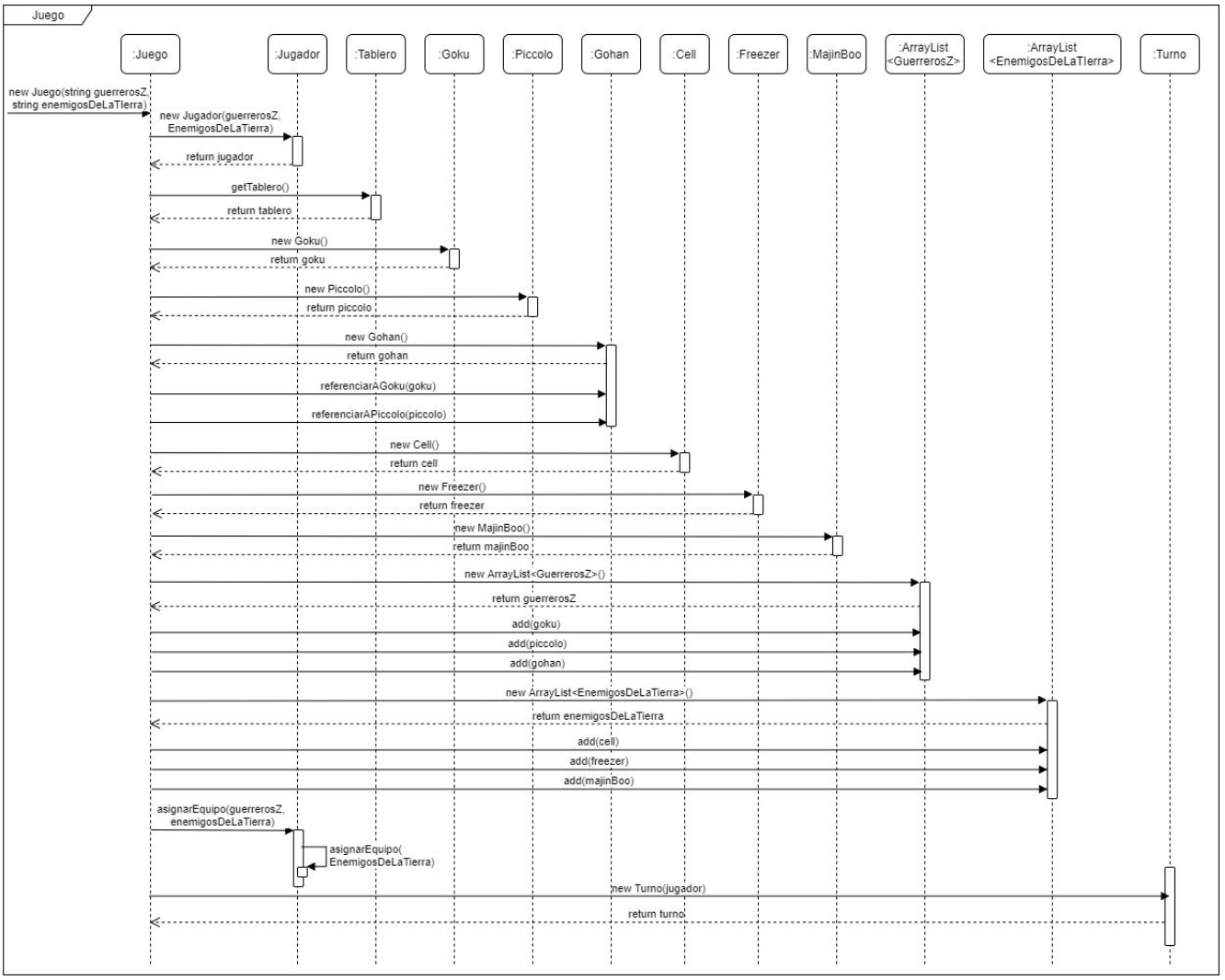
También nos pareció importante diagramar la relación entre los consumibles y los personajes, que son comparables en el sentido de que ambos pueden ubicarse en un casillero. Por lo tanto ambos implementan la interfaz IUbicable.



Los diagramas previamente expuestos grafican de manera generalizada (para los distintos personajes, sus diferentes estados y las interfaces de equipo) la estructura del modelo pensado.

Diagramas de secuencia

Nos pareció interesante realizar un diagrama del método que inicializa el Juego, ya que en él participan varias de las clases de nuestro modelo, y además sirve para exponer la estructura pensada para resolver el TP.



También realizamos un diagrama de secuencia correspondiente al método empleado para mover a los personajes por el tablero. Nos pareció importante ya que deben interactuar varias clases para ello. En pocas palabras, la secuencia consiste en obtener el casillero en el que se encuentra el personaje y desasignarle el personaje, obtener el casillero al que se quiere desplazar el y asignarle el personaje, tomar el consumible que puede o no encontrarse en él y aplicar su efecto sobre el personaje.

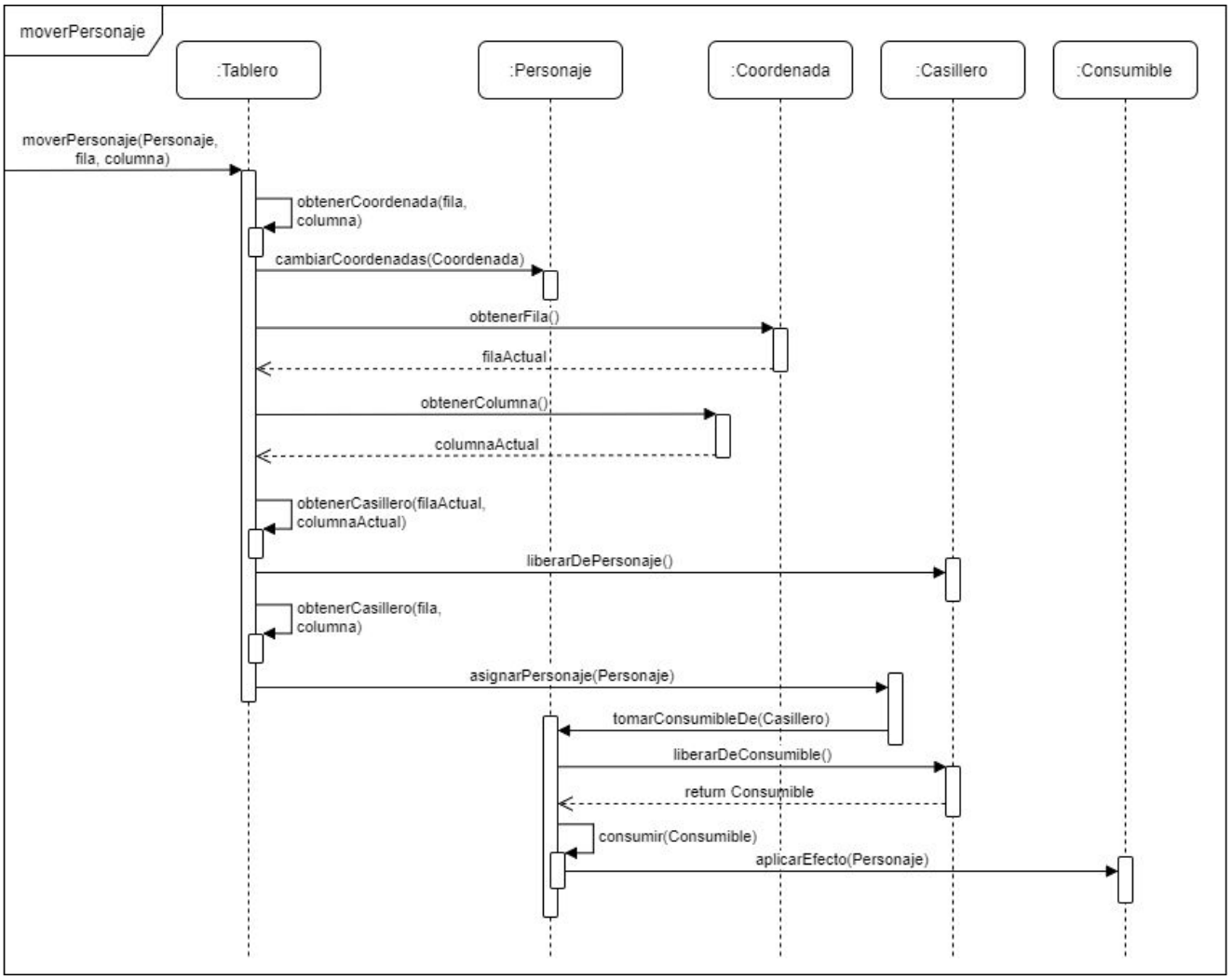
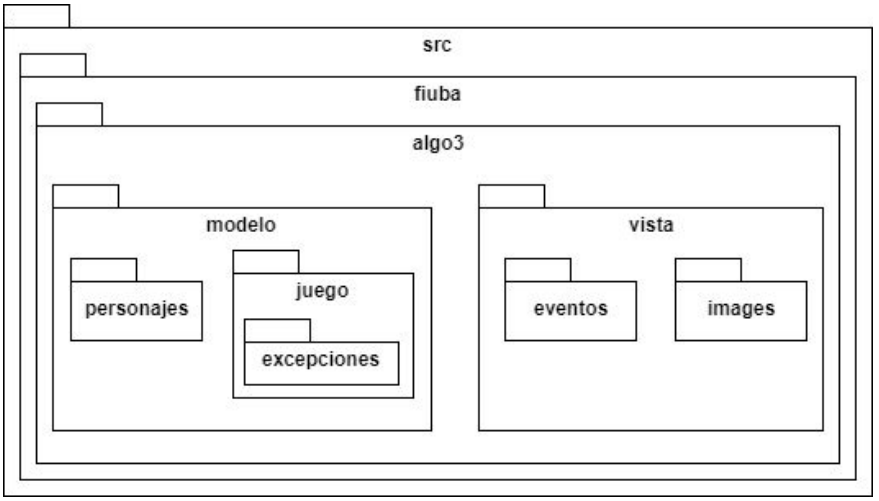


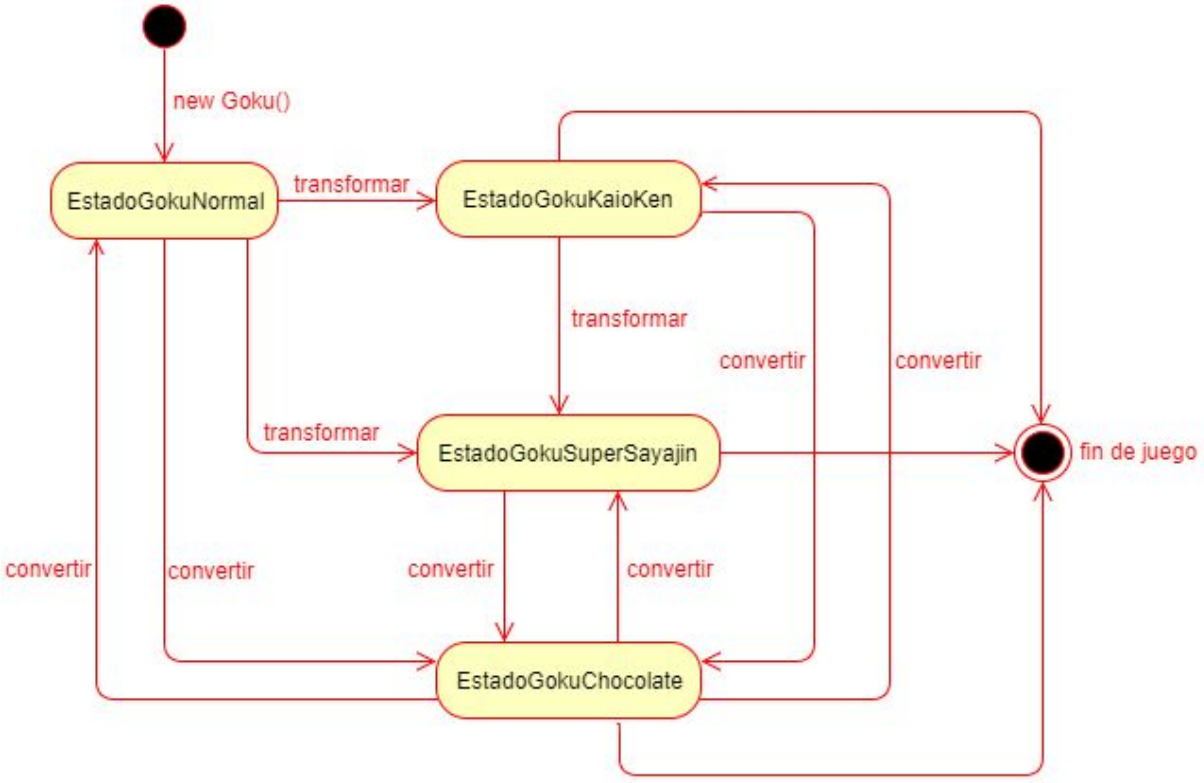
Diagrama de paquetes

A continuación incluimos un diagrama de para mostrar el acoplamiento de nuestro trabajo:

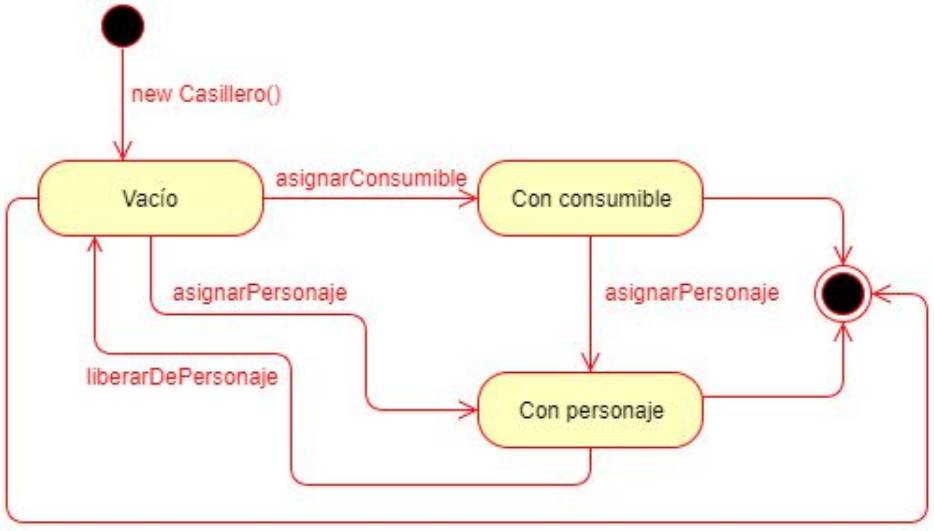


Diagramas de estado

El diagrama a continuación representa los posibles cambios de estado de Goku a lo largo del juego, incluyendo su estado normal, sus dos transformaciones y el estado de chocolate. El caso es análogo al de los personajes Gohan y Piccolo, y similar en el de Cell, Freezer y Majin Boo exceptuando el estado de chocolate.



Agregamos también un diagrama de estados del Casillero, y sus posibles cambios a lo largo del juego.



Detalles de implementación

A continuación se describen algunos de los casos que presentaron complicaciones a la hora de realizar el TP.

Ataque

Uno de los casos más problemáticos con los que tuvimos que trabajar fue el del ataque entre personajes.

Para resolver el problema del ataque entre personajes del mismo equipo, modelamos las interfaces GuerrerosZ y EnemigosDeLaTierra, y sobrecargamos el método ***atacar*** en personaje para que en cada caso reciba un GuerreroZ o un EnemigosDeLaTierra, y responda de modo acorde, ya sea lanzando una excepción o bien efectuando el ataque.

En este último caso, el personaje atacante delega la responsabilidad de atacar a su estado, y a su vez el personaje atacado delega la responsabilidad de recibir el ataque y el daño a su estado.

Movimiento

Para el caso del movimiento de personajes por el tablero, dividimos las responsabilidades entre las clases Tablero, Casillero y Personaje.

El Tablero se ocupa de verificar que la fila y coordenada de destino sean válidas, es decir, que estén en el rango de casilleros del tablero de juego, y delega la responsabilidad de asignar el nuevo casillero al personaje a Casillero y Personaje.

El Casillero del que sale el personaje se encarga de liberarse de él, y el Casillero que recibe el personaje se encarga de guardarlo, verificando si existe un consumible, en cuyo caso se lo transfiere al personaje.

El Personaje se ocupa de asignarse sus nuevas coordenadas luego de verificar que el casillero no esté ocupado.

Conversión en chocolate

Para los ataques especiales de los personajes, creamos el método ***realizarAtaqueEspecial*** en las interfaces GuerrerosZ y EnemigosDeLaTierra, implementado en cada uno de los estados de cada Personaje. En la interfaz de estado de cada personaje creamos también un método con el nombre del ataque especial (por ejemplo, la interfaz EstadoGoku tiene un método kamehameha). Entonces, la implementación del método ***realizarAtaqueEspecial*** en los estados de los personajes consiste en invocar al método con el nombre del ataque especial. Este último verifica las condiciones necesarias y luego invoca al método del personaje oponente que corresponda para recibir el daño.

En particular, el ataque especial de Majin Boo es la conversión del oponente en chocolate. La interfaz EstadoMajinBoo tiene un método ***convertirEnChocolate***, que es implementado en EstadoMajinBooNormal, EstadoMajinBooMalo y EstadoMajinBooOriginal, donde se verifica que el ki sea suficiente, y luego invoca al método ***convertirseEnChocolate*** en el personaje oponente. Este método fue creado en la interfaz GuerrerosZ e implementado por Goku, Gohan y Piccolo, y la implementación consiste en invocar al método convertir, donde se instancia un nuevo estado de chocolate, que tiene un número de turnos correspondiente a los turnos en los que el personaje estará inhabilitado.

Segunda transformación de Gohan

Otro caso problemático fue el de la segunda transformación de Gohan, ya que una de las condiciones para que pudiera darse es que sus compañeros tengan un porcentaje de vida menor al 30%. Para resolverlo, creamos una referencia a Piccolo y una a Goku dentro de Gohan, y al momento de realizar la segunda transformación, además de verificar el ki se verifica el porcentaje de vida de los compañeros.

Excepciones

Se crearon las excepciones a continuación para casos inválidos de jugada.

ExceptionAtaqueAMismoEquipo: es lanzada cuando un personaje intenta atacar a otro de su mismo equipo.

ExceptionNoHayPersonajeEnElCasilleroSeleccionado: es lanzada cuando se intenta atacar a un casillero que no contiene ningún personaje.

ExceptionCantidadDeCasillerosSuperaVelocidad: es lanzada cuando un personaje intenta desplazarse más casilleros que los que le permite su estado en un momento determinado.

ExceptionCasilleroOcupado: es lanzada cuando un personaje intenta ubicarse en un casillero que ya está ocupado por otro personaje.

ExceptionElPersonajeNoLePertenece: es lanzada cuando un personaje no pertenece al equipo que está intentando realizar una jugada, o al equipo enemigo al que se quiere atacar.

ExceptionNoAlcanzaAlOponente: es lanzada cuando un personaje intenta realizar un ataque que supera su distancia de ataque permitida por el estado que le corresponde en un momento particular.

ExceptionFueraDeRango: es lanzada al intentar mover un personaje a un casillero que no pertenece al tablero.

ExceptionGuerreroZConvertidoEnChocolate: es lanzada al intentar realizar una acción con un personaje que se encuentra en estado de chocolate.