**FIUBA - 75.07**

**Algoritmos y programación III**

*Trabajo práctico 2: AlgoPoly*

2do cuatrimestre, 2017

(trabajo grupal de 4 integrantes)

Alumnos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Padrón** | **Mail** |
| Dan Tchechenistky | 100117 | dtcheche@gmail.com |
| Andres Visciglio | 100118 | atvisciglio@gmail.com |
| Tomás Sabao | 99437 | tomalon95@gmail.com |
| Luciano Ortiz | 100323 | lucianortiz97@gmail.com |

***Fecha de entrega final***: 30 de Noviembre de 2017

***Tutor***: Pablo Suárez, Matias Leal Bazterrica

***Comentarios***:

## 

## Modelo de dominio:

El programa está compuesto por total de X clases, con el fin de modelar el juego Monopoly. El siguiente listado menciona y detalla la responsabilidad de cada una de ellas.

* Carcel: Encargada de modificar el movimiento del jugador y de cobrar la fianza.
* Quini6: Clase cuya responsabilidad es premiar al jugador dependiendo de sus recompensas anteriores.
* Policia: Su responsabilidad es enviar al Jugador a la Carcel.
* Caible: Clase abstracta cuyo fin es agrupar Carcel, Quini6, Policia, entre otros y unificar la recepcion de un determinado mensaje, en este caso accionar( ).
* Movimiento Normal Y Carcel encapsulan la movilidad del jugador, de esta forma no es necesario que el jugador conozca la movilidad dentro del tablero.
* RetrocesoDinamico: Su responsabilidad es retroceder una cantidad determinada de casilleros al jugador que caiga en él, dependiendo de lo que saco en sus dados.
* AvanceDinamico: Su responsabilidad es avanzar una cantidad determinada de casilleros al jugador que caiga en él, dependiendo de lo que saco en sus dados.
* BarrioNormal: Clase que representa a aquellas propiedades que se encuentran divididas en Norte y Sur.
* BarrioEspecial: Clase que representa a las demás propiedades que no son companias y que no pueden construir hoteles
* Compania: Clase que agrupa a los distintos servicios Aysa, Edesur,Subtes y Trenes, los cuales tienen mecánicas especiales para el cobro de alquiler y no pueden realizar construcciones.
* Tablero: Clase que representa el tablero de juego. Todos los posibles casilleros ya se encuentran inicializados en el mismo
* EstadoConstruccion: Representa a las construcciones que pueden realizarse en cada propiedad. Se encarga de devolver el alquiler correcto de la propiedad según el número de construcciones realizadas.

## Detalles de implementación:

### Casilleros:

Al momento de plantear la implementación de los casilleros surgió la idea de hacer una clase abstracta Caible que tenga como atributo una posición en el tablero y un método que la accione. De esta clase heredarian entonces los diferentes tipos de casilleros. Por un lado estarían los casilleros que el jugador puede comprar, estos se representan con una clase abstracta Propiedad. Esta contiene los atributos generales a todas las propiedades. Y de ella heredan Barrio y Compañía. Por otro lado se encuentran los casilleros que no pueden ser comprados por el jugador como lo son por ejemplo Carcel, Policia, Avance Dinámico… Todos estos son representados por clases individuales ya que su lógica es independiente.

### Estados de propiedad

Para determinar el comportamiento de las propiedades que poseen o no un dueño, decidimos aplicar el state pattern. Se creó una clase abstracta “EstadoDuenio” de la cual se producen otras dos clases “ConDuenio” y “SinDuenio”. Dado que el comportamiento de las propiedades se encuentra determinado por si tienen o no un propietario, haciendo que al caer los jugadores se mande un mensaje directamente al estado de la propiedad nos ahorra la utilización de numerosos if, haciendo que el código sea más claro y legible, y al mismo tiempo nos otorga más flexibilidad ante cambios en los requisitos propuestos.

### Quini 6

Para la implementación del quini 6 utilizamos una clase abstracta premio, de la cual obtuvimos los 3 premios que pueden ocurrir en el transcurso de una partida: Premio50K

Premio30K y PremioNulo. El jugador guarda una instancia premio, y al momento de caer en Quini6 se cobra el premio actual. Una vez cobrado el premio, el mismo instancia un nuevo premio que corresponde al que le sigue( Premio 30k para Premio50K, PremioNulo para Premio30K) el cual es guardado por el jugador, reemplazando a su predecesor. Una vez que se llega al premio nulo, no se modifica màs el premio.

### Avance y retroceso dinámico:

El avance y retroceso dinamico son muy similares. Al caer un jugador en este casillero se determina cuantas posiciones debera avanzar/retroceder mediante if’s, luego cuentan con un método que hará respectivamente avanzar/retroceder esa cantidad de casilleros al jugador.

Estados de construccion

Se trata de una clase abstracta que modela las construcciones que pueden realizarse en las distintas propiedades. En función de la construcción que representen, cada una devuelve el actual costo de la renta que le corresponde al nivel de construcción. Todas las propiedades arrancan con un estado que representa una construcción nula, y si se cumplen los requisitos, al momento de construir se instancia un nuevo estado que representa la siguiente construcción que puede realizarse.

## Excepciones:

Se implementaron las siguientes excepciones para cubrir posibles errores; MontoInsuficienteException, FianzaInhabilitadaException, JugadorEnBancarrotaExcepcion, RequisitosInsuficientesException

Como es de esperar en el desarrollo del juego, existe la posibilidad de que el jugador no tenga el dinero suficiente para pagar lo que él desea. Es por eso que MontoInsuficienteException aparece en esos casos, para evitar que se realicen operaciones invalidas y perjudiquen la integridad del juego.

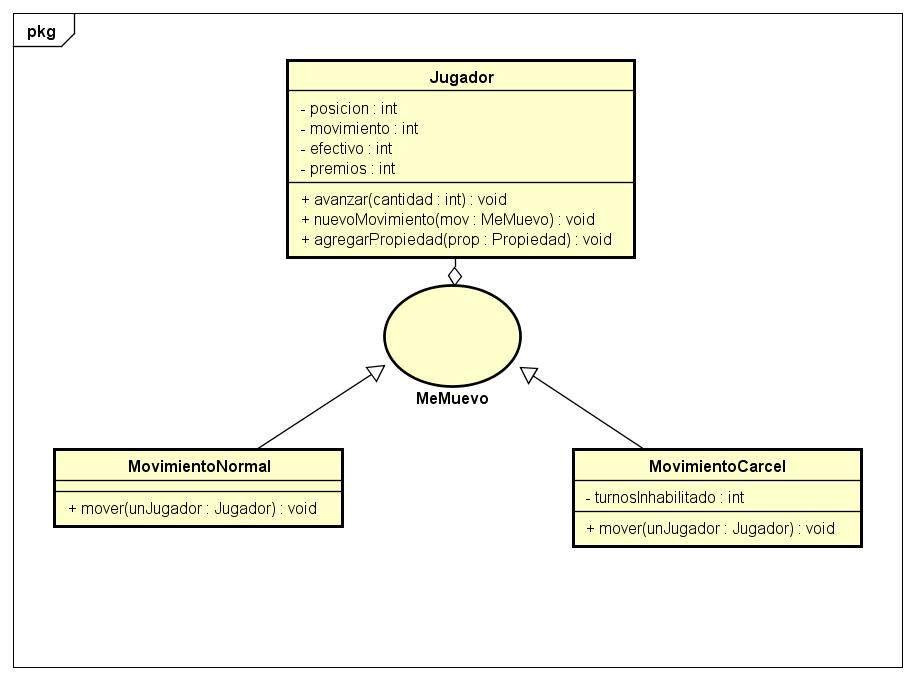
Por otro lado tenemos FianzaInhabilitadaException, la cual surge en los casos en que el jugador intenta pagar la fianza no habiendo dejado pasar un turno obligatorio en la Carcel. De esta forma se respetan las reglas establecidas por el juego.

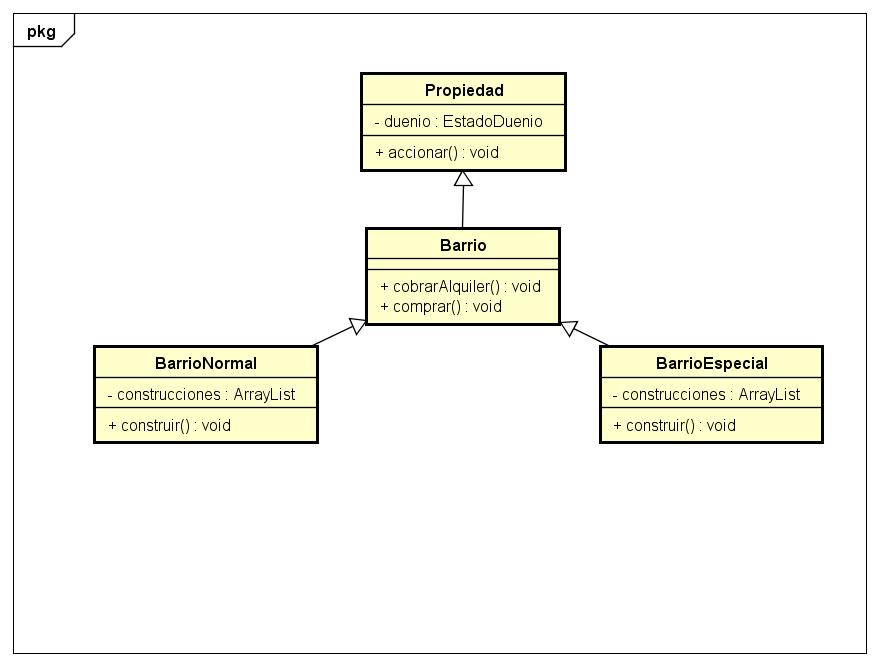
El momento en que un jugador queda eliminado del juego es el momento en que èste queda en bancarrota. JugadorEnBancarrotaExcepcion aparece en el momento en que se le intenta cobrar a un jugador, y el monto a cobrar es superior a lo que puede llegar a pagar usando todos sus recursos.

RequisitosInsuficientesException es una excepcion creada para la realizaciòn de las construcciones en los distintos barrios, y como su nombre lo indica se aplica cuando el jugador no posee los requisitos suficientes para comenzar dicha construcción. Aquellos barrios que poseen tanto una región norte como una región sur tienen como requisito para poder empezar a construir, ambas regiones deben estar en manos de un mismo jugador. Suponiendo el caso de que un jugador trate de construir sin cumplir este requisito, ésta excepción serà lanzada.

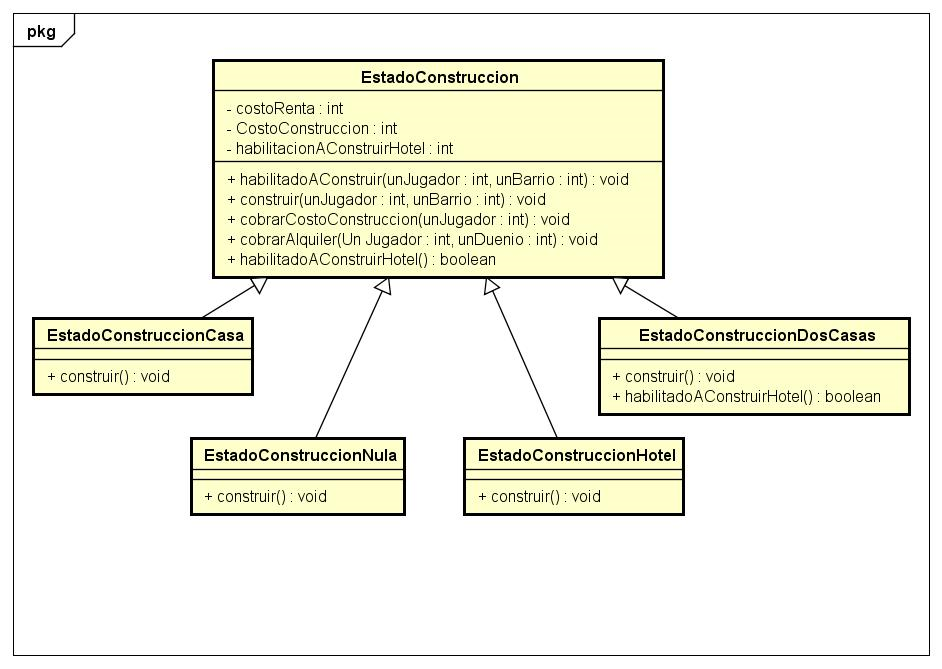
## Diagramas de clase

Diagrama que muestra el movimiento del jugador a traves de una interfaz *MeMuevo*.

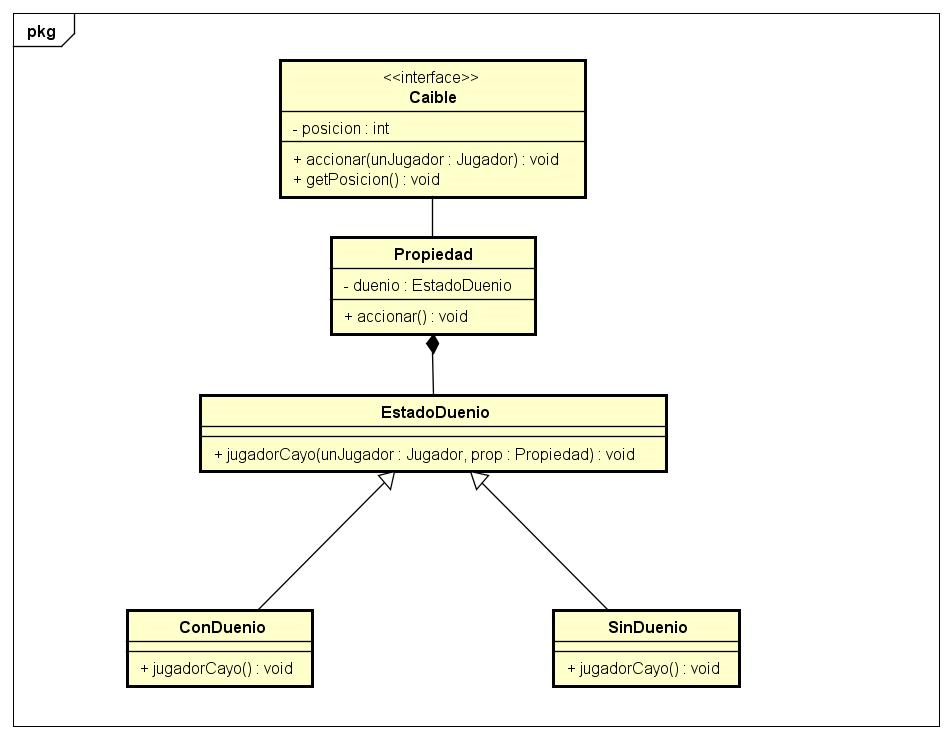


Jerarquia del esquema Barrio - Propiedad

EstadoConstruccion y sus derivados.

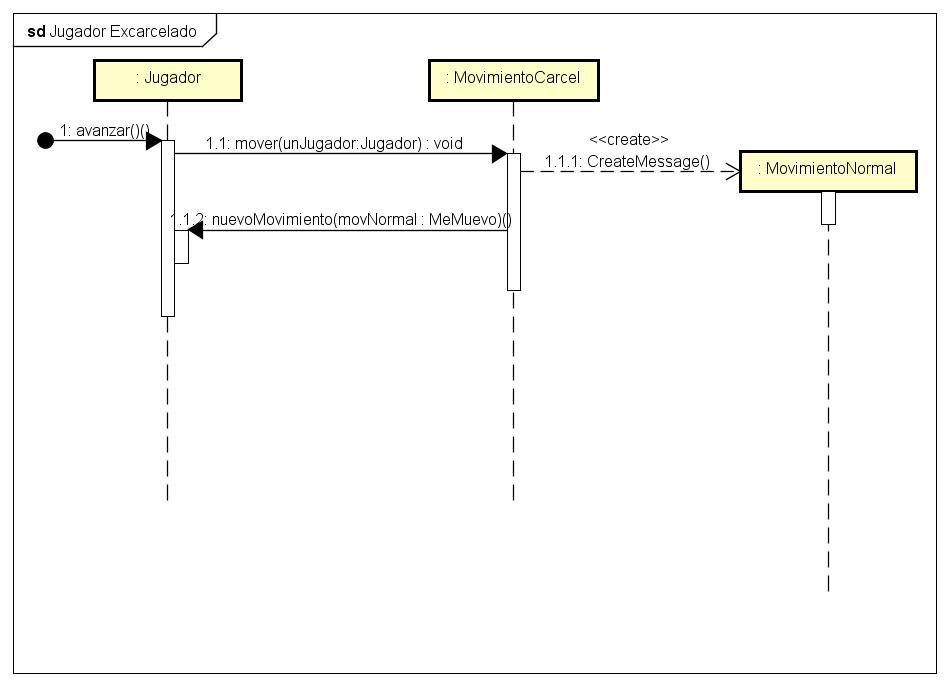


EstadoDuenio y su relacion con Propiedades

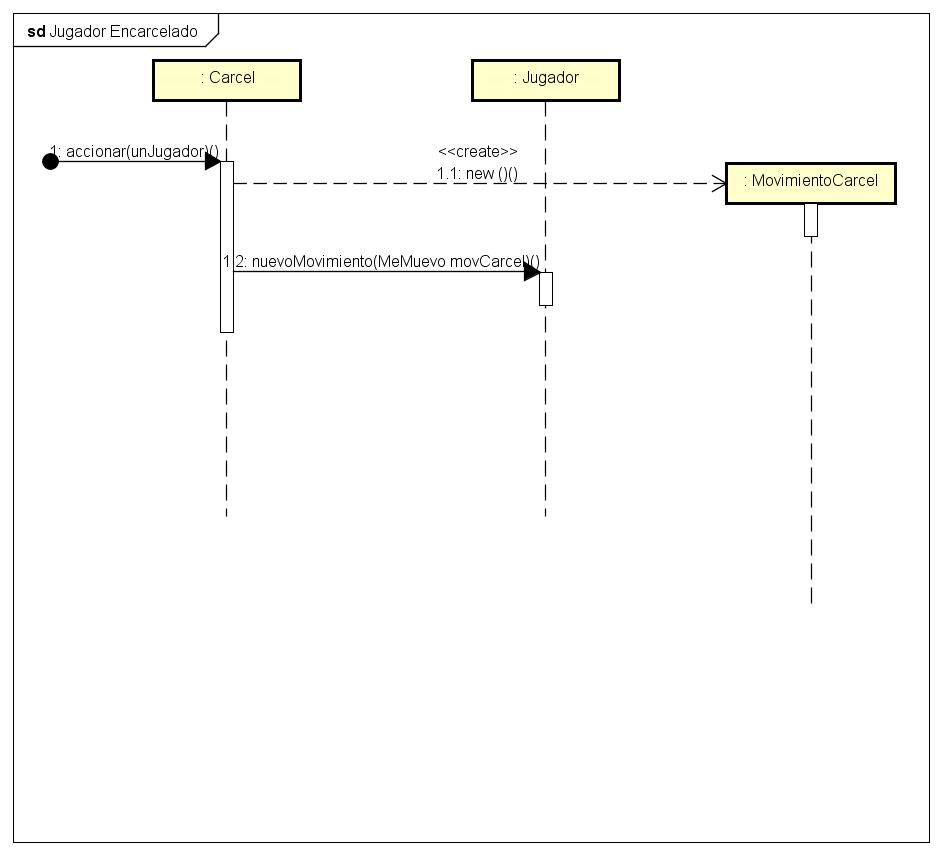


## Diagrama de secuencia

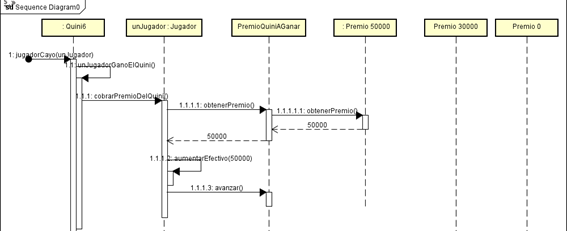
Luego de 4 turnos en la Cárcel, el jugador recupera su movilidad en el juego

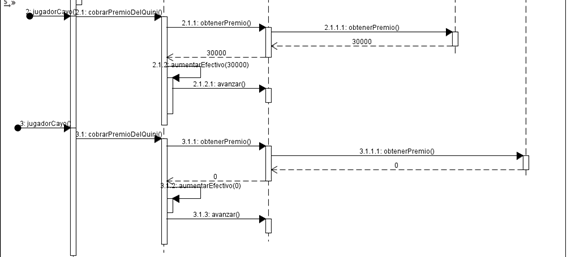


El Jugador es encarcelado y no puede avanzar

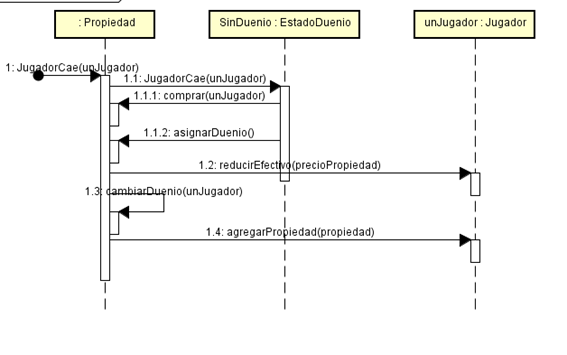


Jugador cae en quini6 3 veces

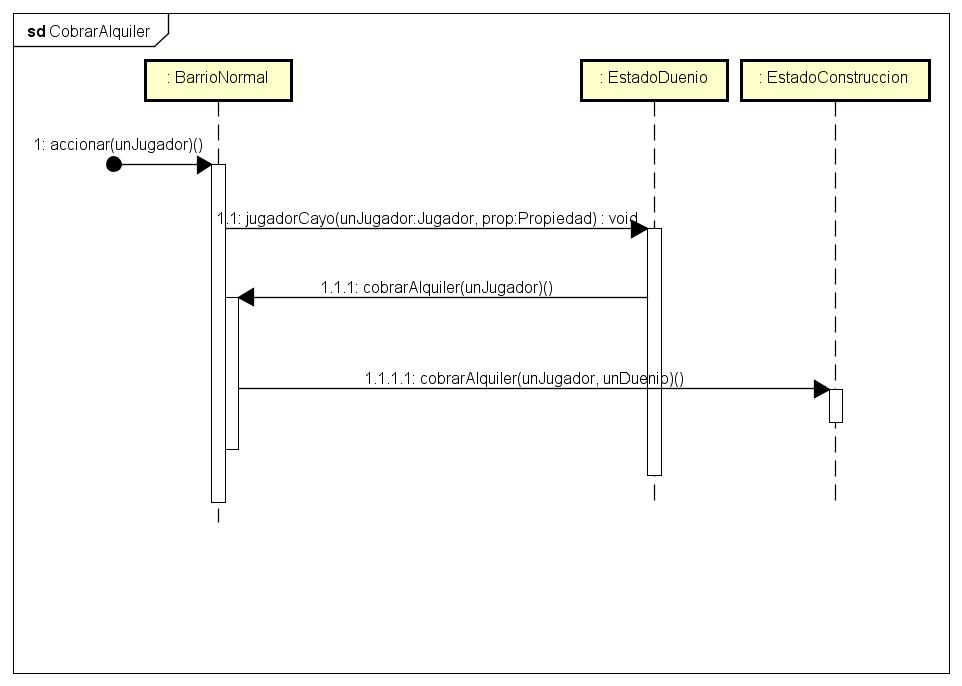




Jugador cae en una propiedad sin dueño y la compra. Él pasa a ser el dueño



Paso a paso, como se le cobra el alquiler a un jugador que cae en una propiedad ajena.



En el siguiente esquema se puede observar cómo funciona el chequeo de la habilitacion para la construccion de un Hotel, en un BarrioNormal.

